

ORDERED COMBINATORY ALGEBRAS AND REALIZABILITY

WALTER FERRER SANTOS, JONAS FREY, MAURICIO GUILLERMO, OCTAVIO MALHERBE,
AND ALEXANDRE MIQUEL

ABSTRACT. We consider different classes of combinatory structures related to Krivine realizability. We show, in the precise sense that they give rise to the same class of triposes, that they are equivalent for the purpose of modeling higher-order logic. We center our attentions in the role of a special kind of Ordered Combinatory Algebras— that we call the *Krivine ordered combinatory algebras* ($^{\mathcal{K}}OCAs$)— that we propose as the foundational pillars for the categorical perspective of Krivine’s classical realizability as presented by Streicher in [23].

Our procedure is the following: we show that each of the considered combinatory structures gives rise to an indexed preorder, and describe a way to transform the different structures into each other that preserves the associated indexed preorders up to equivalence. Since all structures give rise to the same indexed preorders, we only prove that they are triposes once: for the class of $^{\mathcal{K}}OCAs$.

We finish showing that in $^{\mathcal{K}}OCAs$, one can define realizability in every higher-order language and in particular in higher-order arithmetic.

CONTENTS

1. Introduction	2
2. Streicher’s Abstract Krivine Structures	2
2.1. Realizability lattices	3
2.2. The push map in a realizability lattice	3
2.3. Abstract Krivine structures	4
3. Implicative and Krivine ordered combinatory algebras	7
3.1. Ordered combinatory algebras	7
3.2. Implicative ordered combinatory algebras	8
3.3. Krivine ordered combinatory algebras	9
4. Indexed preorders and triposes	9
4.1. Preorders, meet semi-lattices and Heyting preorders	9
4.2. Preorders associated to \mathcal{AKS} s, $OCAs$ and $^{\mathcal{I}}OCAs$	10
4.3. Indexed preorders and indexed meet-semi-lattices	12
4.4. Triposes	13
5. Constructing triposes from ordered structures	14
5.1. From \mathcal{AKS} s to indexed preorders	14
5.2. From $OCAs$ to indexed meet-semilattices	15
5.3. From $^{\mathcal{I}}OCAs$ to triposes	15
5.4. From \mathcal{AKS} s to $^{\mathcal{K}}OCAs$	16
5.5. From $^{\mathcal{K}}OCAs$ to \mathcal{AKS} s	18
6. Internal realizability in $^{\mathcal{K}}OCAs$	20
References	23

1. INTRODUCTION

Classical realizability was introduced in the mid 90's by Krivine as a complete reformulation of the principles of Kleene's (intuitionistic) realizability (see [10]), to take into account the connection between control operators and classical reasoning discovered by Griffin (in [3]). Initially developed in the framework of classical second-order Peano arithmetic (see [11]), classical realizability was quickly extended to Zermelo-Fraenkel set theory in [13] using model-theoretic constructions reminiscent both to the construction of generic extensions in forcing and to the construction of intuitionistic realizability models of intuitionistic set theories, see [21], [1], [20]. In particular, Krivine showed in [14] how to interpret the (classical) axiom of dependent choices in this framework. More recently, he also showed in [12] how to combine classical realizability with the method of forcing (in the sense of Cohen), in the very spirit of iterated forcing.

Actually, Krivine's realizability (particularly in its most recent developments) was mostly developed regardless to the long-standing tradition of intuitionistic realizability. And, as mentioned by Streicher in [23], it was difficult to see how Krivine's work could fit into the structural approach to realizability as initiated by Hyland in [6] and fully described in [24]. One problem comes from the fact that the only realizability topos that fulfils classical logic is the one based on the trivial partial combinatory algebra (\mathcal{PCA}) and thus, equivalent to **Set**, a fact that suggested for a long time that realizability and classical logic were incompatible.

To resolve this paradox, Streicher proposed in [23] a categorical model for Krivine's realizability, still using the standard method that consists to combine the construction of a realizability tripos with the well known tripos-to-topos construction (see [24]). However, Streicher's construction of the realizability tripos departs from the standard construction from a \mathcal{PCA} in several aspects.

First, Streicher does not use a \mathcal{PCA} , but a particular form of ordered combinatory algebra (\mathcal{OCA}), that is built from an abstract Krivine structure (\mathcal{AKS}) that provides the computational ingredients of Krivine realizability.

Second, the elements of the considered \mathcal{OCA} (induced by the underlying \mathcal{AKS}) are not used as realizers, but directly as truth values, using the fact that the considered \mathcal{OCA} has a meet-semilattice structure. In this way, Streicher can skip the step that consists to take the powerset to define truth values, and more generally relations (as one would do if working with a \mathcal{PCA}).

A third ingredient of Streicher's construction is the introduction of a specific notion of filter, to distinguish the truth values that actually capture the notion of truth/provability. In practice, this filter is naturally defined from the pole of the \mathcal{AKS} and the corresponding notion of proof-like terms. Two warnings here concerning notations: following the usual trends we use sometimes the expression quasi-proof following the original french *quasi-preuve* instead of proof-like term; also the reader should be aware that the notion of filter used in this context is due to Hofstra (see [4]) and is different from the usual one.

In this paper, we revisit Streicher's work by showing that his construction can be performed working directly from a particular form of \mathcal{OCA} , which we call ${}^{\mathcal{K}}\mathcal{OCA}$, whose elements can be indifferently used as realizers (or conditions) and as truth values, similarly to the elements of a complete Boolean algebra in forcing. In particular, it should be clear to the reader that complete Boolean algebras are particular cases of ${}^{\mathcal{K}}\mathcal{OCAs}$, and that in this case, the general construction presented here amounts to the standard construction of a Boolean tripos. So that the concept of ${}^{\mathcal{K}}\mathcal{OCA}$ can be seen as the common denominator between classical realizability and Cohen forcing. Moreover, to make more striking the comparison with the standard approach of categorical models of realizability, we actually present the tripos construction starting from a slightly more general structure of ${}^{\mathcal{I}}\mathcal{OCA}$ that does not assume anything about the logic being classical.

2. STREICHER'S ABSTRACT KRIVINE STRUCTURES

As motivation for the introduction of the concept of *Krivine ordered combinatory algebras* (${}^{\mathcal{K}}\mathcal{OCAs}$), we recapitulate the definitions and basic ideas in [23], regarding the notion of *Abstract Krivine Structures* \mathcal{AKS} .

These ideas were introduced by J.L. Krivine and reformulated categorically by T. Streicher –see [15] and [23] respectively–.

2.1. Realizability lattices.

Definition 2.1. A realizability lattice –abbreviated as \mathcal{RL} – consists of:

- (1) A triple (Λ, Π, \perp) where Λ and Π are sets, of terms and stacks respectively, and $\perp \subseteq \Lambda \times \Pi$ is a subset (relation) ($\Lambda \times \Pi$ is the set of processes and its elements are written (t, π) or $(t, \pi) = t \star \pi$, moreover if $t \star \pi \in \perp$, we write $t \perp \pi$, i.e. “ t is orthogonal to π or t realizes π ”). If $P \subseteq \Pi$ and $t \perp \pi$ for all $\pi \in P$ we say that “ t realizes P ” and write $t \perp P$.
- (2) Define the following maps:

$$(\)^\perp : \mathcal{P}(\Lambda) \longrightarrow \mathcal{P}(\Pi)$$

$$\Lambda \supseteq L \longmapsto L^\perp = \{\pi \in \Pi \mid \forall t \in L, t \star \pi \in \perp\} = \{\pi \in \Pi \mid L \times \{\pi\} \subseteq \perp\} \subseteq \Pi;$$

$${}^\perp(\) : \mathcal{P}(\Pi) \longrightarrow \mathcal{P}(\Lambda)$$

$$\Pi \supseteq P \longmapsto {}^\perp P = \{t \in \Lambda \mid \forall \pi \in P, t \star \pi \in \perp\} = \{t \in \Lambda \mid \{t\} \times P \subseteq \perp\} \subseteq \Lambda.$$

- (3) Define the following sets:

$$\mathcal{P}_\perp(\Lambda) = \{L \subseteq \Lambda : {}^\perp(L^\perp) = L\} \subseteq \mathcal{P}(\Lambda) \quad , \quad \mathcal{P}_\perp(\Pi) = \{P \subseteq \Pi : ({}^\perp P)^\perp = P\} \subseteq \mathcal{P}(\Pi).$$

Remark 2.2. (1) The maps $L \rightarrow L^\perp$ and $P \rightarrow {}^\perp P$ are antitonic with respect to the order given by the inclusion of sets and for $P_i \subseteq \Pi$, $L_i \subseteq \Lambda$, $i \in I$ we have:

$${}^\perp(\bigcap_{i \in I} P_i) \supseteq \bigcup_{i \in I} {}^\perp P_i, \quad {}^\perp(\bigcup_{i \in I} P_i) = \bigcap_{i \in I} {}^\perp P_i; \quad (\bigcap_{i \in I} L_i)^\perp \supseteq \bigcup_{i \in I} L_i^\perp, \quad (\bigcup_{i \in I} L_i)^\perp = \bigcap_{i \in I} L_i^\perp.$$

- (2) For an arbitrary $L \in \mathcal{P}(\Lambda)$ and $P \in \mathcal{P}(\Pi)$, one has that ${}^\perp(L^\perp) \supseteq L$ and $({}^\perp P)^\perp \supseteq P$. For an arbitrary $L \in \mathcal{P}(\Lambda)$ and $P \in \mathcal{P}(\Pi)$, one has that $({}^\perp(L^\perp))^\perp = L^\perp$ and ${}^\perp({}^\perp P)^\perp = {}^\perp P$.
- (3) The maps $(\)^\perp : \mathcal{P}(\Lambda) \rightarrow \mathcal{P}(\Pi)$ and ${}^\perp(\) : \mathcal{P}(\Pi) \rightarrow \mathcal{P}(\Lambda)$ when restricted respectively to $\mathcal{P}_\perp(\Lambda)$ and $\mathcal{P}_\perp(\Pi)$ are order reversing isomorphisms inverse to each other. An \mathcal{RL} satisfies two strong completion properties. If \mathcal{X} is a subset of $\mathcal{X} \subseteq \mathcal{P}_\perp(\Pi)$, define:

$$\sup(\mathcal{X}) = \left({}^\perp \left(\bigcup \{P : P \in \mathcal{X}\} \right) \right)^\perp, \quad \inf(\mathcal{X}) = \bigcap \{P : P \in \mathcal{X}\}.$$

In particular, $\sup(\mathcal{X})$ and $\inf(\mathcal{X})$ are the supremum and infimum of the set \mathcal{X} in $\mathcal{P}_\perp(\Pi)$ with respect to the order given by the inclusion of sets. Moreover with respect to the order given by the inclusion, Λ^\perp and Π ; ${}^\perp \Pi$ and Λ are the minimal and maximal elements of $\mathcal{P}_\perp(\Pi)$ and $\mathcal{P}_\perp(\Lambda)$ respectively.

The only relevant structure at this point is the lattice structure in the sets $\mathcal{P}_\perp(\Lambda)$ and $\mathcal{P}_\perp(\Pi)$, where we take the (set theoretical) inclusion as the order the intersection as “meet” and the union followed by taking double orthogonals as “join”.

2.2. The push map in a realizability lattice. In this section we add a *push map* to a realizability lattice, thus introducing the first elements of a *calculus* into our structure.

Definition 2.3. (1) A map $\text{push}(t, \pi) : (t, \pi) \mapsto t \cdot \pi : \Lambda \times \Pi \rightarrow \Pi$ defined in a realizability lattice (Λ, Π, \perp) , will be called a *push map*.

- (2) For an \mathcal{RL} with a push map, and for $L \subseteq \Lambda$, $P \subseteq \Pi$ we define,

$$(L, P) \mapsto L \rightsquigarrow P : \mathcal{P}(\Lambda) \times \mathcal{P}(\Pi) \rightarrow \mathcal{P}(\Pi)$$

with $L \rightsquigarrow P = \{\pi \in \Pi : L \cdot \pi \subseteq P\} \subseteq \Pi$ –called the right conductor of L into P . We consider also $(L, P) \mapsto L \cdot P : \mathcal{P}(\Lambda) \times \mathcal{P}(\Pi) \rightarrow \mathcal{P}(\Pi)$ where: $L \cdot P = \{t \cdot \pi : t \in L, \pi \in P\}$.

Remark 2.4. Clearly, for L and P as above: $L \cdot P \subseteq Q$ if and only if $P \subseteq L \rightsquigarrow Q$.

The above constructions of $L \rightsquigarrow P$ and $L.P$ combined with the operators $(\)^\perp$ and $^\perp(\)$ yield natural binary operations in $\mathcal{P}_\perp(\Pi)$ that are basic ingredients of the *OCA* associated to the *AKS à la Streicher*.

We define the following binary operations between subsets of Π .

Definition 2.5. *Let $P, Q \subseteq \Pi$ then:*

$$(2.1) \quad P \circ Q := {}^\perp Q \rightsquigarrow P \subseteq P \circ_\perp Q := ({}^\perp({}^\perp Q \rightsquigarrow P))^\perp,$$

$$(2.2) \quad P \Rightarrow Q := {}^\perp P \cdot Q \subseteq P \Rightarrow_\perp Q := ({}^\perp({}^\perp P \cdot Q))^\perp \in \mathcal{P}_\perp(\Pi).$$

Once the above definitions are established, we can deduce a crucial “half adjunction property” relating the operations \circ_\perp and \Rightarrow_\perp in $\mathcal{P}_\perp(\Pi)$.

Theorem 2.6. [Half adjunction property] *Assume that $P, Q, R \in \mathcal{P}_\perp(\Pi)$. If $Q \Rightarrow_\perp R \subseteq P$, then $R \subseteq P \circ_\perp Q$. In particular: $P \subseteq (Q \Rightarrow_\perp P) \circ_\perp Q$.*

Proof. The following inclusions are equivalent: $Q \Rightarrow_\perp R \subseteq P$, $({}^\perp({}^\perp Q \cdot R))^\perp \subseteq P$, ${}^\perp Q \cdot R \subseteq P$ and $R \subseteq {}^\perp Q \rightsquigarrow P$. The last inclusion implies that $R \subseteq ({}^\perp({}^\perp Q \rightsquigarrow P))^\perp = P \circ_\perp Q$. \square

2.3. Abstract Krivine structures. Next, –compare with [23]– we complete the process of adding a calculus to a realizability lattice to obtain the concept of *Abstract Krivine Structure* abbreviated as *AKS*. For that, we introduce the usual application map for terms, a store map from stacks to terms, the combinators κ, s , and a distinguished term cc that is a realizer of Peirce’s law. We introduce also a set of terms that we call quasi-proofs and assume that the three combinators above are quasi proofs.

Definition 2.7. *An Abstract Krivine Structure (frequently written as \mathcal{K}) consists of the following elements:*

- (1) *A realizability lattice with a push $(\Lambda, \Pi, \perp, \text{push})$,*
- (2) *Functions*
 - (a) *$\text{app} : \Lambda \times \Lambda \rightarrow \Lambda$ is a function: $(t, u) \mapsto \text{app}(t, u) = tu$,*
 - (b) *$\text{store} : \Pi \rightarrow \Lambda$ is a function: $\pi \mapsto \text{store}(\pi) = k_\pi$,*
- (3) *A set $\text{QP} \subseteq \Lambda$ of “quasi-proofs”, which is closed under application,*
- (4) *Elementary combinators $\kappa, s, cc \in \text{QP}$.*
- (5) *The above elements are subject to the following axioms.*
 - (S1) *If $t \perp s \cdot \pi$, then $ts \perp \pi$.*
 - (S2) *If $t \perp \pi$, then for all $s \in \Lambda$ we have that $\kappa \perp t \cdot s \cdot \pi$.*
 - (S3) *If $tu(su) \perp \pi$, then $s \perp t \cdot s \cdot u \cdot \pi$.*
 - (S4) *If $t \perp k_\pi \cdot \pi$, then $cc \perp t \cdot \pi$.*
 - (S5) *If $t \perp \pi$, then for all $\pi' \in \Pi$ we have that $k_\pi \perp t \cdot \pi'$.*

Here –and in the rest of this paper– the product-like operations will not be associative and we assume that when parenthesis are omitted, we associate to the left.

The elements of the structure above, named as:

$$\text{store} : \pi \mapsto k_\pi : \Pi \rightarrow \Lambda \quad \text{and} \quad cc \in \Lambda,$$

have a very special role in the sense they can be used to make the realizability theory *classical* as cc realizes Peirce’s law. In this sense in the presence of the mentioned elements and the corresponding axioms (S4) and (S5), the *AKS*– is *classical*.

Definition 2.8. *For a general *AKS* we introduce the following definitions:*

- (1) *For $L, M \subseteq \Lambda$ we define $L \Rightarrow M = \{t \in \Lambda : tL \subseteq M\}$,*
- (2) *For $P, Q \subseteq \Pi$, $P \diamond Q := ({}^\perp P)({}^\perp Q)^\perp \in \mathcal{P}_\perp(\Pi)$.*
- (3) *$I = s\kappa\kappa$, $B = s(\kappa s)\kappa$, $E = s(\kappa I) \in \text{QP}$.*

Lemma 2.9. *In an *AKS*, for $P, Q \in \mathcal{P}_\perp(\Pi)$, we have that condition (S1) in Definition 2.7, (5) implies any of the two equivalent conditions below.*

- (1) $P \circ_\perp Q \subseteq ({}^\perp P {}^\perp Q)^\perp = P \diamond Q$

(2) If $t \perp P$ and $s \perp Q$, then $ts \perp P \circ_{\perp} Q$.

Proof. It is evident that the two conditions above are equivalent. Assuming (S1), we want to prove that for all P, Q then: $\{\pi \in \Pi : {}^{\perp}Q.\pi \subseteq P\} \subseteq ({}^{\perp}P^{\perp}Q)^{\perp}$.

In other words we want to show that if $\pi \in \Pi$ is such that ${}^{\perp}Q.\pi \subseteq P$ then, for all $s \perp P, t \perp Q$ we have that $st \perp \pi$. It is clear that from the hypothesis ${}^{\perp}Q.\pi \subseteq P$ and $s \perp P, t \perp Q$, that $s \perp t.\pi$ and in this case the original condition (S1) implies that $st \perp \pi$. \square

Next, we deduce some consequences or equivalent formulations of the basic axioms for an \mathcal{AKS} in terms of elements of $\mathcal{P}_{\perp}(\Pi)$ and the operations $\circ_{\perp}, \rightarrow_{\perp}, \diamond$.

Lemma 2.10. For $P, Q, R \in \mathcal{P}_{\perp}(\Pi)$, $t, u, v \in \Lambda$, and $\pi \in \Pi$, we have:

- (1) $t \perp (P \Rightarrow_{\perp} Q) \Rightarrow_{\perp} R$ if and only if $t \perp (P \Rightarrow Q) \Rightarrow R$. Also if $t \perp P \Rightarrow_{\perp} (Q \Rightarrow_{\perp} R)$ then $t \perp P \Rightarrow (Q \Rightarrow R)$.
- (2) $t \perp P \Rightarrow_{\perp} Q, u \perp P$ implies $tu \perp Q$;
- (3) $\kappa \perp P \Rightarrow Q \Rightarrow P$;
- (4) $s \perp (P \Rightarrow Q \Rightarrow R) \Rightarrow (P \Rightarrow Q) \Rightarrow P \Rightarrow R$;
- (5) $cc \perp ((P \Rightarrow_{\perp} Q) \Rightarrow_{\perp} P) \Rightarrow_{\perp} P$ or equivalently $cc \perp ((P \Rightarrow Q) \Rightarrow P) \Rightarrow P$;
- (6) $t \perp \pi$ implies $\mathbf{I} \perp t.\pi$;
- (7) $\mathbf{I} \perp P \Rightarrow_{\perp} P$;
- (8) $t \perp uv.\pi$ implies $\mathbf{B} \perp t.u.v.\pi$;
- (9) $\mathbf{B} \perp (Q \Rightarrow R) \Rightarrow (P \Rightarrow Q) \Rightarrow P \Rightarrow R$;
- (10) $tu \perp \pi$ implies $\mathbf{E}t \perp u.\pi$.

Proof. Ad 1 – This follows immediately from the fact that taking orthogonals three times is the same than taking them once.

Ad 2 – Take $t \perp (P \Rightarrow_{\perp} Q)$ and $u \in {}^{\perp}P$ and we want to prove that $tu \perp Q$ or equivalently that ${}^{\perp}(P \Rightarrow_{\perp} Q)({}^{\perp}P) \subseteq {}^{\perp}Q$ or that $(P \Rightarrow_{\perp} Q) \diamond P \supseteq Q$. We have (Lemma 2.9) that $(P \Rightarrow_{\perp} Q) \diamond P \supseteq (P \Rightarrow_{\perp} Q) \circ_{\perp} P \supseteq Q$. See Theorem 2.6 for the last inequality.

Ad 3 – Let $t \in {}^{\perp}P, u \in {}^{\perp}Q, \pi \in P$. We have to show that $\kappa \perp t.u.\pi$. By (S2), it is sufficient to show $t \perp \pi$, which follows from the definition of ${}^{\perp}P$.

Ad 4 – Let $t \in {}^{\perp}(P \Rightarrow Q \Rightarrow R), u \in {}^{\perp}(P \Rightarrow Q), v \in {}^{\perp}P, \pi \in R$. Using 2 we can deduce $tv(uv) \in {}^{\perp}R$ and thus $tv(uv) \perp \pi$. Axiom (S3) implies $\mathbf{S} \perp t.u.v.\pi$, as required.

Ad 5 – Let $t \in {}^{\perp}((P \Rightarrow_{\perp} Q) \Rightarrow_{\perp} P)$ and $\pi \in P$. We have to show that $cc \perp t.\pi$, and by (S4) it is sufficient to show that $t \perp k_{\pi}.\pi$. This would follow from $k_{\pi} \in {}^{\perp}(P \Rightarrow_{\perp} Q)$, so it remains to prove the latter.

Let $u \in {}^{\perp}P, \pi' \in Q$. We have to show that $k_{\pi} \perp u.\pi'$, and by (S5) it is sufficient to show that $u \perp \pi$, which is true since $u \in {}^{\perp}P$ by assumption.

Ad 6 – $t \perp \pi$ implies $\mathbf{I} \perp t.\pi$. Indeed, $t \perp \pi \Rightarrow \kappa \perp t \cdot (\kappa t) \cdot \pi \Rightarrow \kappa t \perp \kappa t \cdot \pi \Rightarrow \kappa t(\kappa t) \perp \pi \Rightarrow \mathbf{S} \perp \kappa \cdot \kappa \cdot t \cdot \pi \Rightarrow \mathbf{I} = \mathbf{SKK} \perp t \cdot \pi$.

Ad 7 – It is clear that the assertion $\mathbf{I} \perp P \Rightarrow_{\perp} P$ is another formulation of 6.

Ad 10 – The following chain of implications proves that $tu \perp \pi$ implies $\mathbf{E}t := \mathbf{S}(\kappa(\mathbf{SKK}))t \perp u.\pi$.

$$\begin{aligned} tu \perp \pi &\Rightarrow \mathbf{I} = \mathbf{SKK} \perp tu \cdot \pi \Rightarrow \kappa \perp \mathbf{SKK} \cdot u \cdot tu \cdot \pi \Rightarrow \kappa(\mathbf{SKK})u(tu) \perp \pi \\ &\Rightarrow \kappa(\mathbf{SKK})u(tu) \perp \pi \Rightarrow \mathbf{S} \perp (\kappa(\mathbf{SKK})) \cdot t \cdot u \cdot \pi \Rightarrow \mathbf{S}(\kappa(\mathbf{SKK}))t \perp u \cdot \pi. \end{aligned}$$

In the first implication we used the definition of \mathbf{I} in the second the definition of κ , in the third we used (S1) in the fourth the definition of \mathbf{S} and in the last one, we used property (S1) again. \square

Remark 2.11. Clauses 2-5 resemble the Hilbert style axiomatization of the implicational fragment of classical propositional logic.

Using this analogy, it is easy to show the following.

Assume that $\varphi[X_1, \dots, X_n]$ is a propositional formula built up from propositional variables X_1, \dots, X_n and implication. For arbitrary subsets $P_1, \dots, P_n \subseteq \Pi$ denote by $\varphi[P_1, \dots, P_n] \subseteq \Pi$ the evaluation of $\varphi[X_1, \dots, X_n]$ where P_1, \dots, P_n are substituted for the variables, and implication is interpreted by the operation \Rightarrow from Definition 2.5. If $\varphi[X_1, \dots, X_n]$ is provable in the Hilbert calculus, then $\varphi[P_1, \dots, P_n]$ contains a quasi-proof, namely the element of \mathbf{QP} obtained by evaluating the proof term of $\varphi[X_1, \dots, X_n]$ in Λ .

Next we list some properties satisfied by any combinator that satisfies the $s\eta$ rule (i.e. the rule 10) in particular by \mathbb{E} .

Lemma 2.12. (1) If a combinator $\widehat{\mathbb{E}}$ satisfies the $s\eta$ rule –i.e. $ts \perp \pi \Rightarrow \widehat{\mathbb{E}}t \perp s \cdot \pi$ – then it satisfies any of the equivalent assertions that follow.

$$(2.3) \quad \text{If } P, Q \in \mathcal{P}_\perp(\Pi) \text{ then } P \diamond Q \subseteq \{\pi \in \Pi : \widehat{\mathbb{E}}(\perp P) \subseteq \perp(\perp Q \cdot \pi)\} = \{\pi \in \Pi : (\widehat{\mathbb{E}}(\perp P))^\perp \supseteq (\perp Q \cdot \pi)\}.$$

$$(2.4) \quad \text{If } P, Q \in \mathcal{P}_\perp(\Pi) \text{ then } \widehat{\mathbb{E}}(\perp P) \subseteq \perp(\perp Q \cdot (P \diamond Q)).$$

$$(2.5) \quad \text{If } R \subseteq (P \diamond Q), \text{ with } P, Q, R \in \mathcal{P}_\perp(\Pi) \text{ then } \widehat{\mathbb{E}}(\perp P) \subseteq \perp(\perp Q \cdot R).$$

(2) If a combinator $\widehat{\mathbb{E}}$ satisfies the $s\eta$ rule then –in the notations of Definition 2.3–, the following assertions hold.

$$(2.6) \quad \text{If } P, Q \in \mathcal{P}_\perp(\Pi) \text{ then } \widehat{\mathbb{E}}(\perp(\perp P \cdot Q)) \subseteq \perp(\perp P \cdot Q) \text{ or equivalently } \widehat{\mathbb{E}}(\perp(P \Rightarrow_\perp Q)) \subseteq \perp(P \Rightarrow_\perp Q).$$

$$(2.7) \quad (t(\perp P))^\perp \subseteq \{\pi \in \Pi : (\widehat{\mathbb{E}}t)^\perp \supseteq (\perp P \cdot \pi)\} \subseteq (\perp\{\pi \in \Pi : (\widehat{\mathbb{E}}t)^\perp \supseteq (\perp P \cdot \pi)\})^\perp = (\widehat{\mathbb{E}}t)^\perp \circ_\perp P.$$

$$(2.8) \quad \text{If } P, Q \in \mathcal{P}_\perp(\Pi) \text{ then } (P \diamond Q) \subseteq (\widehat{\mathbb{E}}(\perp P))^\perp \circ_\perp Q.$$

(3) If $P \in \mathcal{P}_\perp(\Pi)$ and $\widehat{\mathbb{E}}$ are as above, then: $(\widehat{\mathbb{E}}(\perp P))^\perp \subseteq (\widehat{\mathbb{E}}\widehat{\mathbb{E}})^\perp \circ_\perp P$.

Proof. (1) It is clear that the inclusions (2.3), (2.4), (2.5) are all equivalent.

Now we prove that if $\forall s, t, \pi, ts \perp \pi \Rightarrow \widehat{\mathbb{E}}t \perp s \cdot \pi$ then the inclusion (2.3) holds.

Take $\pi \in P \diamond Q$. This means that for all $t \perp P, s \perp Q, ts \perp \pi$ and then $\widehat{\mathbb{E}}t \perp s \cdot \pi$. This means that the inclusion (2.3) holds.

(2) The chain of inclusions below, show that if the combinator $\widehat{\mathbb{E}}$ satisfies the $s\eta$ rule, then (2.8) holds. The other results are proved similarly.

$$P \diamond Q \subseteq \{\pi \in \Pi : (\widehat{\mathbb{E}}(\perp P))^\perp \supseteq (\perp Q \cdot \pi)\} \subseteq (\perp\{\pi \in \Pi : (\widehat{\mathbb{E}}(\perp P))^\perp \supseteq (\perp Q \cdot \pi)\})^\perp = (\widehat{\mathbb{E}}(\perp P))^\perp \circ_\perp Q.$$

Note that the first inclusion is just (2.3).

(3) The proof follows by substitution of t by $\widehat{\mathbb{E}}t$ in (2.7). □

The theorem that follows recovers (partially) “the other half” of the half adjunction property of Theorem 2.6.

Theorem 2.13. Let $P, Q, R \in \mathcal{P}_\perp(\Pi)$. If

$$R \subseteq P \circ_\perp Q \text{ then } (Q \Rightarrow_\perp R) \subseteq (\mathbb{E}(\perp P))^\perp \subseteq (\mathbb{E}\mathbb{E})^\perp \circ_\perp P.$$

Proof. As $R \subseteq P \circ_\perp Q \subseteq P \diamond Q = (\perp P^\perp Q)^\perp$ –see Lemma 2.9, we have that $\perp Q \cdot R \subseteq \perp Q \cdot (\perp P^\perp Q)^\perp$ and $\perp(\perp Q \cdot R) \supseteq \perp(\perp Q \cdot (\perp P^\perp Q)^\perp)$. By equation (2.4) we have that $\mathbb{E}(\perp P) \subseteq \perp(\perp Q \cdot (\perp P^\perp Q)^\perp) \subseteq \perp(\perp Q \cdot R) = \perp(Q \Rightarrow_\perp R)$. Taking orthogonals we obtain the first inclusion. The other inclusion is just Lemma 2.12, (3), since by Lemma 2.10, (10) \mathbb{E} satisfies the $s\eta$ -rule. □

Definition 2.14. In a \mathcal{AKS} as above, the combinator $\mathbb{E}\mathbb{E} \in \mathbf{QP}$ is called an adjunction.

3. IMPLICATIVE AND KRIVINE ORDERED COMBINATORY ALGEBRAS

In [23], Streicher presented a construction of an *ordered combinatory algebra* (\mathcal{OCA}) (see [24, Section 1.8]) out of an \mathcal{AKS} , from which he constructed a tripos whose predicates are functions with values in the \mathcal{OCA} . This construction does not give rise to a tripos in general, but only for some \mathcal{OCA} s – in particular for those induced by \mathcal{AKS} ’s. The notion of *implicative ordered combinatory algebra* abbreviated as \mathcal{IOCA} , is an axiomatization of the additional structure that is necessary on an \mathcal{OCA} to guarantee that the induced indexed preorder is a tripos. The tripos will be classical in case the \mathcal{IOCA} has an additional combinator, called c , that realizes Peirce’s law.

In this section we fix our attention on *implicative ordered combinatory algebras*: \mathcal{IOCA} s and the modification consisting in adding the combinator c that produces a *Krivine ordered combinatory algebra*: $\kappa\mathcal{OCA}$. The main features added to the usual structure of an ordered combinatory algebra –compare with [4]– are the following: a) we assume the existence of a distinguished element, that we call *adjunction*; b) we assume that the \mathcal{IOCA} is inf –complete; c) we have an implication mapping denoted as \rightarrow . These additions are present in the \mathcal{OCA} s that come from \mathcal{AKS} s, and will be crucial ingredients in the construction of the associated tripos, that we build up directly from the \mathcal{IOCA} –compare with [23]–. See for example [9] and [24] for the standard approach to the subject.

3.1. Ordered combinatory algebras.

Definition 3.1. (1) An ordered combinatory algebra (\mathcal{OCA}) is a quintuple $\mathcal{A} = (A, \leq, \text{app}, k, s)$ – written frequently as A – where (A, \leq) is a partial order,

$$\text{app} : A \times A \rightarrow A, \quad (a, b) \mapsto ab$$

is a monotone function, and k, s are elements of A satisfying

- (a) $kab \leq a$
- (b) $sabc \leq ac(bc)$

for all $a, b, c \in A$.

- (2) A filter in an \mathcal{OCA} –called A – is a subset $\Phi \subseteq A$ which contains s and k and is closed under application. A pair (\mathcal{A}, Φ) is called a *filtered \mathcal{OCA}* .

Remark 3.2. Here –and in the rest of this paper– the product-like operations will not be associative and we assume that when parenthesis are omitted, we associate to the left.

In what follows, we will recall how to program directly in this \mathcal{OCA} , using the standard codifications in the combinatory algebras.

Definition 3.3. Let A be an \mathcal{OCA} and take a denumerable set of variables: $\mathcal{V} = \{x_1, x_2, \dots\}$ and consider $A(\mathcal{V})$ –called the set of terms in A – that is the set of formal expressions given by the following grammar: $p_1, p_2 ::= a \mid x \mid p_1 p_2$ where $a \in A$ and $x \in \mathcal{V}$. As usual $A(x_1, \dots, x_k)$ is the set of terms in A containing only the variables x_1, \dots, x_k . One can naturally extend the order in A to an order in $A(\mathcal{V})$ in such a way that: if $p_1 R p_2, q_1 R q_2 \in A(\mathcal{V})$ then $p_1 q_1 R p_2 q_2$ and if $p_1, p_2 \in A(\mathcal{V})$ then $k p_1 p_2 R p_1$ and if $p_1, p_2, p_3 \in A(\mathcal{V})$ then $s p_1 p_2 p_3 R p_1 p_3 (p_2 p_3)$.

If we define an equivalence relation \equiv_R on $A(\mathcal{V})$ as: $p \equiv_R q$ iff $p R q$ and $q R p$, the order R can be factored to a partial order in the quotient $A[\mathcal{V}] := A(\mathcal{V}) / \equiv_R$ –called also the *set of terms in A* –.

If p_1, p'_1, p_2, p'_2 are terms such that $p_1 \equiv_R p'_1$ and $p_2 \equiv_R p'_2$ then $p_1 p_2 \equiv_R p'_1 p'_2$. Hence, the application can be defined in the quotient and is written as $p_1 \star p_2$.

Hence, $(A[\mathcal{V}], R, \star)$ is an \mathcal{OCA} and (A, \leq, app) is a sub- \mathcal{OCA} of $(A[\mathcal{V}], R, \star)$.

It is customary to denote the relation R as \leq and the operation \star as \circ or as the concatenation of the factors. In this situation we say that $(A[\mathcal{V}], \leq, \circ)$ is an extension of (A, \leq, \circ) .

The following result is well known.

Theorem 3.4 (Combinatory completeness). *For any finite set of variables $\{x_1, \dots, x_k, y\}$, there is a function $\lambda^*y : A[x_1, \dots, x_k, y] \rightarrow A[x_1, \dots, x_k]$ satisfying the following property:*

$$\text{If } t \in A[x_1, \dots, x_k, y], \text{ and } u \in A[x_1, \dots, x_k] \text{ then } (\lambda^*y(t)) \circ u \leq t\{y := u\}.$$

Moreover if $X \subseteq A$ is an arbitrary subset and t is a term with all its coefficients in X , then $\lambda^*y(t)$ is a term with all its coefficients in $\langle X \rangle$, the closure of X by application. In particular if all the coefficients of t are in the filter Φ , then $\lambda^*y(t)$ is a polynomial with all the coefficients in Φ . Occasionally we write $\lambda^*y(t) = \lambda^*y.t$.

Proof. The function λ^*y is defined recursively: i) If $y \neq x$, then $\lambda^*y(x) := kx$; ii) $\lambda^*y(y) := s k k$; iii) if p, q are polynomials, then: $\lambda^*y(pq) := s(\lambda^*y(p))(\lambda^*y(q))$. From the fact that $\langle X \rangle$ contains k, s and it is closed under applications, we deduce the condition on the coefficients of $\lambda^*y(t)$. \square

We use combinatory completeness to define some combinators that we will use later.

Definition 3.5. *Let A be an \mathcal{OCA} , we can define the following combinators or combinatorial functions that are elements of Φ , or functions with codomain and domain Φ .*

$$\begin{aligned} b &= \lambda^*x \lambda^*y \lambda^*z(x(yz)) & i &= \lambda^*x(x) & c &= \lambda^*x \lambda^*y \lambda^*z(zxy) & w &= \lambda^*x \lambda^*y(xy y) \\ t &= \lambda^*x \lambda^*y(x) & f &= \lambda^*x \lambda^*y(y) & p &= \lambda^*x \lambda^*y \lambda^*z(zxy) & p_0 &= \lambda^*x(x t) & p_1 &= \lambda^*x(x f) \\ a &: \Phi \times \Phi \rightarrow \Phi & a(r, s) &= \lambda^*x(p(r x)(s x)) & d &: \Phi \rightarrow \Phi & d(f) &= \lambda^*x(f(p_0 x)(p_1 x)). \end{aligned}$$

Lemma 3.6. *If A is an \mathcal{OCA} , the above definitions ensure that:*

$$\begin{aligned} babc &\leq a(bc); \quad ia \leq a; \quad cab c \leq acb; \quad wab \leq abb; \quad p_0(pab) \leq a; \quad p_1(pab) \leq b; \\ rc &\leq a, \quad sc \leq b \Rightarrow a(r, s)c \leq pab; \quad d(f)\ell \leq f(p_0\ell)(p_1\ell). \end{aligned}$$

for all $a, b, c, \ell \in A$.

3.2. Implicative ordered combinatory algebras.

Definition 3.7. *An implicative ordered combinatory algebra $-a^I\mathcal{OCA}-$, consists of an inf-complete partially ordered set (A, \leq) equipped with:*

(1) *binary operations*

$$\text{app} : A \times A \rightarrow A, \quad (a, b) \mapsto ab$$

called application, monotone in both arguments, and

$$\text{imp} : A^{\text{op}} \times A \rightarrow A, \quad (a, b) \mapsto a \rightarrow b$$

called implication, antitonic in the first argument and monotone in the second;

(2) *a subset $\Phi \subseteq A$ (called filter) which is closed under application;*

(3) *distinguished elements $s, k, e \in \Phi$*

such that the following holds for all $a, b, c \in A$.

$$\begin{aligned} (\text{PK}) \quad kab &\leq a \\ (\text{PS}) \quad sab c &\leq ac(bc) \\ (\text{PA}) \quad a \leq b \rightarrow c &\Rightarrow ab \leq c \\ (\text{PE}) \quad ab \leq c &\Rightarrow ea \leq b \rightarrow c \end{aligned}$$

3.3. Krivine ordered combinatory algebras.

Definition 3.8. A Krivine ordered combinatory algebra $\neg a^{\kappa\text{OCA}}\neg$, consists of an $\mathcal{I}\text{OCA}$ equipped with a distinguished element $c \in \Phi$ such that for all $a, b \in A$,

$$(PC) \quad c \leq ((a \rightarrow b) \rightarrow a) \rightarrow a.$$

Next we show that in the definition of a κOCA (and also of a $\mathcal{I}\text{OCA}$) some of its elements are superfluous and can be obtained from the others. Here we present a minimal setup for the concept.

Definition 3.9. A quadruple $\mathcal{Q} = (A, \leq, \rightarrow, \Phi)$ where

- (1) The relation \leq is a partial order in A with the property that each $X \subseteq A$ has an infimum.
- (2) The map $\rightarrow: A \times A \rightarrow A$ –implication– is Antone in the first variable and monotone in the second.
- (3) $\Phi \subset A$ –a filter– is a subset of A .

is said to be proper if Φ satisfies the following conditions.

Define:

- (1) a map $\text{app}: A \times A \rightarrow A$ –called the application– given for $a, b \in A$ as:

$$\text{app}(a, b) = ab := \inf\{c : a \leq (b \rightarrow c)\}.$$

- (2) $k := \inf\{a \rightarrow (b \rightarrow a) : a, b \in A\}.$
- (3) $s := \inf\{a \rightarrow (b \rightarrow (c \rightarrow (ac)(bc))) : a, b, c \in A\}.$
- (4) $e := \inf\{a \rightarrow (b \rightarrow ab) : a, b \in A\}.$
- (5) $c := \inf\{((a \rightarrow b) \rightarrow a) \rightarrow a : a, b \in A\}.$

then

- (1) The set Φ is closed under the application: $\text{app}(\Phi, \Phi) \subset \Phi$.
- (2) The elements $k, s, e, c \in \Phi$.

Theorem 3.10. If $\mathcal{Q} = (A, \leq, \rightarrow, \Phi)$ is proper (Definition 3.9), then $\mathcal{A}_{\mathcal{Q}} = (A, \leq, \rightarrow, \Phi, \text{app}, k, s, e, c)$ is a κOCA .

Proof. To prove the half adjunction property, assume that for $a, b, c \in A$, $a \leq (b \rightarrow c)$ as ab is the infimum of the elements c with the above property, it is clear that in this situation $ab \leq c$ (condition (PA) of Definition 3.8: the “half adjunction property”). The fact that the application $(a, b) \mapsto ab$ is monotone in both variables follows directly from the definition by using the monotony properties of the implication. If $a, b, c \in A$ are such that $ab \leq c$, by definition we know that $e \leq a \rightarrow (b \rightarrow ab)$ and then that $e \leq a \rightarrow (b \rightarrow c)$. Applying the half adjunction property ((PA) in the previous definition) we obtain that $e a \leq (b \rightarrow c)$. The satisfaction by k, s of the required properties follows by a direct application of (PA) and the condition for c is evidently satisfied. \square

4. INDEXED PREORDERS AND TRIPOSES

4.1. Preorders, meet semi-lattices and Heyting preorders.

Definition 4.1. We denote by **Ord** the category of preorders and monotone maps. A preorder (D, \leq) is a set D with a reflexive and transitive relation \leq . A monotone map between the preorders (D, \leq) and (E, \leq) is a function $f: D \rightarrow E$ such that $d \leq d'$ implies $f(d) \leq f(d')$ for all $d, d' \in D$. If $d \leq d'$ and $d' \leq d$, we say that d and d' are isomorphic, and write $d \cong d'$.

Definition 4.2. Let (C, \leq) and (D, \leq) be two preorders.

- (1) For monotone maps $f, g: (C, \leq) \rightarrow (D, \leq)$, we define $f \leq g : \Leftrightarrow \forall d \in D. f(d) \leq g(d)$ and say that f and g are isomorphic (written $f \cong g$) if $f \leq g$ and $g \leq f$.
- (2) A monotone map $f: (C, \leq) \rightarrow (D, \leq)$ is called an equivalence, if there exists a monotone map $g: (D, \leq) \rightarrow (C, \leq)$ such that $g \circ f \cong \text{id}_D$, and $f \circ g \cong \text{id}_E$ and g is called a weak inverse of f . In this situation we say that (C, \leq) and (D, \leq) are equivalent (written $(D, \leq) \simeq (E, \leq)$).

- (3) Given monotone maps $f : (C, \leq) \rightarrow (D, \leq)$, $g : (D, \leq) \rightarrow (C, \leq)$, we say that ‘ f is left adjoint to g ’, or ‘ g is right adjoint to f ’, and write $f \dashv g$, if $\text{id}_C \leq g \circ f$ and $f \circ g \leq \text{id}_D$.

Remark 4.3. The following assertions are easy to prove.

- (1) A monotone map $f : (C, \leq) \rightarrow (D, \leq)$ is an equivalence if and only if it is order reflecting and essentially surjective, i.e.
 - (a) $\forall c, c' \in D. f(c) \leq f(c') \Rightarrow c \leq c'$, and
 - (b) $\forall d \in D \exists c \in C. f(c) \cong d$.
- (2) Let $f : (C, \leq) \rightarrow (D, \leq)$, $g : (D, \leq) \rightarrow (C, \leq)$ be monotone maps between preorders.
 - (a) f is left adjoint to g , if and only if $\forall c \in C, d \in D. f(c) \leq d \Leftrightarrow c \leq g(d)$.
 - (b) Adjoints are unique up to isomorphism, i.e. when $f \dashv g$ and $f \dashv g'$, then $g \cong g'$ (and similarly for left adjoints).

Definition 4.4. A meet semi-lattice is a preorder (D, \leq) equipped with a binary operation \wedge and a distinguished element \top such that for all $a, b, c \in D$:

- (1) $a \wedge b \leq a$;
- (2) $a \wedge b \leq b$;
- (3) $c \leq a$ and $c \leq b \Rightarrow c \leq a \wedge b$;
- (4) $a \leq \top$.

Remark 4.5. If (D, \leq) is a meet semi-lattice, then the function $(d, d') \mapsto d \wedge d'$ is a monotone map of type $D \times D \rightarrow D$, which is right adjoint to the diagonal map $\delta : D \rightarrow D \times D, d \mapsto (d, d)$.

Definition 4.6. **SLat** is the category of meet semi-lattices, and meet preserving monotone maps, i.e. monotone maps $f : (D, \leq) \rightarrow (E, \leq)$ such that

- (1) $f(d) \wedge f(d') \cong f(d \wedge d')$ for all $d, d' \in D$
- (2) $f(\top) \cong \top$.

Definition 4.7. We define **HPO**, the category of Heyting preorders and morphisms.

- (1) A Heyting preorder is a meet semi-lattice (A, \leq) with a binary operation $\rightarrow : A \times A \rightarrow A$ (called Heyting implication) satisfying

$$(H) \quad a \wedge b \leq c \quad \text{if and only if} \quad a \leq b \rightarrow c$$
 for all $a, b, c \in A$.
- (2) A morphism of Heyting preorders is a monotone map $f : (A, \leq) \rightarrow (B, \leq)$ such that
 - (a) $f(\top) \cong \top$
 - (b) $f(a \wedge b) \cong f(a) \wedge f(b)$
 - (c) $f(a \rightarrow b) \cong f(a) \rightarrow f(b)$
 for all $a, b \in A$.

Remarks 4.8. (1) The term ‘Heyting preorder’ is not standard, but it is the same as a ‘posetal Cartesian closed category’, or equivalently a preorder whose poset reflection is a ‘Heyting semi-lattice’ [8, Part A1.5].

- (2) A Heyting preorder with finite joins is what is called a Heyting prealgebra, e.g. in [24]. The anti-symmetric version is the well known concept of Heyting algebra.
- (3) To interpret disjunction we also want joins in triposes, but we don’t have to postulate disjunction (and neither \exists), since they can be encoded in terms of the other connectives in second order logic.
- (4) Also, we don’t have to demand Heyting implication to be monotone – it follows from the definition that it is antitonic in the first, and monotone in the second variable.

4.2. Preorders associated to AKSs, OCAs and ${}^I\text{OCAs}$.

AKS

Definition 4.9. Let $\mathcal{K} = (\Lambda, \Pi, \dots)$ be an abstract Krivine structure. We define the relation \sqsubseteq in $\mathcal{P}(\Pi)$ as follows:

$$(4.1) \quad P, Q \in \mathcal{P}(\Pi), \quad P \sqsubseteq Q \quad :\Leftrightarrow \quad \exists t \in \text{QP} \quad t \perp P \Rightarrow Q$$

for $P, Q \in \mathcal{P}(\Pi)$. An element $t \in \Phi$ as above is said to be “a realizer of the relation” $P \sqsubseteq Q$.”

Remark 4.10. Notice that the relation above, could have been defined using the arrow \Rightarrow_{\perp} . Indeed, $t \perp P \Rightarrow Q$ if and only if $t \perp P \Rightarrow_{\perp} Q$ by Lemma 2.10, (1).

Lemma 4.11. Let \mathcal{K} be an abstract Krivine structure, then the relation \sqsubseteq is a preorder on $\mathcal{P}(\Pi)$.

Proof. The combinator \mathbf{I} is a realizer of $P \sqsubseteq P$ for any $P \in \mathcal{P}(\Pi)$, thus \sqsubseteq is reflexive. For transitivity, assume that $P, Q, R \in \mathcal{P}(\Pi)$, and that $t, u \in \text{QP}$ are realizers of $P \sqsubseteq Q$ and $Q \sqsubseteq R$, respectively. Then $\mathbf{B}tu$ is a realizer of $P \sqsubseteq R$. \square

Lemma 4.12. The canonical inclusion $\mathcal{P}_{\perp}(\Pi) \hookrightarrow \mathcal{P}(\Pi)$ is an equivalence of preorders with respect to \sqsubseteq .

Proof. By Remark 4.3 it suffices to show that the inclusion is order reflecting and essentially surjective. Since the order on $\mathcal{P}_{\perp}(\Pi)$ is defined as restriction of the order on $\mathcal{P}(\Pi)$ the first assertion is clear.

To prove that the inclusion is essentially surjective, we show that $P \sqsubseteq (\perp P)^{\perp}$ and $(\perp P)^{\perp} \sqsubseteq P$ for all $P \in \mathcal{P}(\Pi)$. This holds since

$$\mathbf{I} \perp \perp((\perp P)^{\perp}) \cdot P = \perp P \cdot P \quad \text{and} \quad \mathbf{I} \perp \perp P \cdot (\perp P)^{\perp}$$

for all $P \in \Pi$. Both relations are realized by \mathbf{I} as follows directly from Lemma 2.10, (7) applied in the cases of P and $(\perp P)^{\perp}$, respectively. \square

OCA

Definition 4.13. Let (\mathcal{A}, Φ) be a filtered OCA . We define:

(1) The relation \sqsubseteq_{Φ} in A as follows:

$$a \sqsubseteq_{\Phi} b, \text{ if and only if } \exists f \in \Phi : fa \leq b.$$

(2) A map $\wedge : A \times A \rightarrow A$ as $a \wedge b := \text{pab}$ –see Definition 3.5.

(3) An element $\top \in \Phi$

Usually we omit the subscript Φ in the notation of the relation \sqsubseteq_{Φ} and write $a \sqsubseteq b$. An the element f as above is said to be “a realizer of the relation $a \sqsubseteq b$ ” and write this assertion as $f \Vdash a \sqsubseteq b$.

We establish some properties that will be of later use.

Lemma 4.14. If (\mathcal{A}, Φ) is a filtered OCA then in the notations of Definition 3.5 we have that:

- (1) $\text{p}_0 \Vdash a \wedge b \sqsubseteq a$
- (2) $\text{p}_1 \Vdash a \wedge b \sqsubseteq b$
- (3) If $r \Vdash c \sqsubseteq a$ and $s \Vdash c \sqsubseteq b$ then $\mathbf{a}(r, s) \Vdash c \sqsubseteq a \wedge b$
- (4) $\text{kk} \Vdash a \sqsubseteq \top$.

Hence, (A, \wedge, \sqsubseteq) is a meet-semi-lattice.

Proof. All the assertions follow directly from Lemma 3.6. \square

IOCA

Next we show that in the case of the existence of an adjunction, more precise assertions can be proved concerning the meet and the order \sqsubseteq .

Theorem 4.15. If (\mathcal{A}, Φ) is a IOCA then:

- (1) If $a, b \in A$ then $a \sqsubseteq b$ if and only if there is an element $f \in \Phi$ such that $f \leq a \rightarrow b$.

(2) If $a, b, c \in A$:

$$a \wedge b \sqsubseteq c \Leftrightarrow a \sqsubseteq (b \rightarrow c).$$

In other words $(A, \sqsubseteq, \wedge, \rightarrow)$ is a Heyting preorder.

Proof. (1) Assuming that $f \leq a \rightarrow b$ and using the half adjunction property we deduce that $fa \leq b$ i.e. that $a \sqsubseteq b$. In case that $a \sqsubseteq b$, first we deduce that $ga \leq b$ for some $g \in \Phi$. Using the adjunction we deduce that $eg \leq a \rightarrow b$.

(2) To see that the map \rightarrow gives a Heyting implication on (A, \sqsubseteq) , we have to check that

$$a \wedge b \sqsubseteq c \Leftrightarrow a \sqsubseteq (b \rightarrow c)$$

where $(a \wedge b) = \mathbf{p}ab$.

If the left inequality holds, there exists an element $f \in \Phi$ such that $fa \leq b \rightarrow c$, and Definition 3.8, (PA) gives $fab \leq c$. In accordance with Lemma 3.6 there exists a function $d : \Phi \rightarrow \Phi$ such that $d(f)\ell \leq f(\mathbf{p}_0\ell)(\mathbf{p}_1\ell)$ for all $\ell \in A$, and this gives (substituting ℓ by $\mathbf{p}ab$)

$$d(f)(a \wedge b) = d(f)(\mathbf{p}ab) \leq fab \leq c.$$

Conversely, assume that the right hand side holds, i.e. there exists an $f \in \Phi$ such that $f(\mathbf{p}ab) \leq c$. Then we can deduce

$$\begin{array}{c} f(\mathbf{p}ab) \leq c \\ \hline \mathbf{b}f(\mathbf{p}a)b \leq c \\ \hline \mathbf{b}(\mathbf{b}f)\mathbf{p}ab \leq c \\ \hline \mathbf{e}(\mathbf{b}(\mathbf{b}f)\mathbf{p}a) \leq b \rightarrow c \\ \hline \mathbf{b}\mathbf{e}(\mathbf{b}(\mathbf{b}f)\mathbf{p})a \leq b \rightarrow c, \end{array}$$

hence $\mathbf{b}\mathbf{e}(\mathbf{b}(\mathbf{b}f)\mathbf{p})$ is a realizer of $a \sqsubseteq b \rightarrow c$.

□

For future use we prove the following property of the combinator \mathbf{c} in the case that the $\mathcal{I}\mathcal{OCA}$ is equipped with one.

Lemma 4.16. *Assume that the \mathcal{A} is a $\mathcal{I}\mathcal{OCA}$ equipped with an element \mathbf{c} with the property that for if $a, b \in \mathcal{A}$ then, $\mathbf{c} \leq ((a \rightarrow b) \rightarrow a) \rightarrow a$. If $a \in \mathcal{A}$, then $\mathbf{c} \Vdash ((a \rightarrow \perp) \rightarrow \perp) \sqsubseteq a$.*

Proof. For $a \in \mathcal{A}$ we have:

$$\begin{array}{c} \mathbf{c} \leq ((a \rightarrow \perp) \rightarrow a) \rightarrow a \\ \hline \mathbf{c}((a \rightarrow \perp) \rightarrow a) \leq a \\ \hline \mathbf{c}((a \rightarrow \perp) \rightarrow \perp) \leq \mathbf{c}((a \rightarrow \perp) \rightarrow a) \leq a \\ \hline \mathbf{c} \Vdash ((a \rightarrow \perp) \rightarrow \perp) \sqsubseteq a \end{array}$$

□

4.3. Indexed preorders and indexed meet-semi-lattices.

Definition 4.17. (1) An indexed preorder is a functor $\mathcal{D} : \mathbf{Set}^{\text{op}} \rightarrow \mathbf{Ord}$.

(2) An indexed meet-semi-lattice is a functor $\mathcal{A} : \mathbf{Set}^{\text{op}} \rightarrow \mathbf{SLat}$.

(3) An indexed Heyting preorder is a functor $\mathcal{P} : \mathbf{Set}^{\text{op}} \rightarrow \mathbf{HPO}$.

We only present the following definitions in the case of preorders, in the case of indexed meet-semilattices the concepts are similar.

Remarks 4.18. (1) Indexed preorders (in particular triposes, defined below) can be used as categorical models of predicate logic. With this in mind, we often call their elements predicates – more precisely, if \mathcal{D} is an indexed preorder, I is a set, and $\varphi \in \mathcal{D}(I)$, we say that φ is a predicate on I .

- (2) If \mathcal{D} is an indexed preorder and $f : J \rightarrow I$ is a function, applying the functor to f gives us a monotone map $\mathcal{D}(f) : \mathcal{D}(I) \rightarrow \mathcal{D}(J)$. We call this function reindexing along f , and usually abbreviate it by f^* . Thus, if φ is a predicate on I , then its reindexing $f^*(\varphi)$ along f is a predicate on J . Semantically, reindexing corresponds to substitution and context extension.
- (3) There are more general concepts of indexed preorder, one is that of a pseudofunctor of type $\mathbf{Set}^{\text{op}} \rightarrow \mathbf{Ord}$. Another generalization of indexed preorders is to replace \mathbf{Set} by another category. We do not need these levels of generality.
- (4) Preorders are a special case of indexed categories, which are functors $\mathcal{C} : \mathbf{Set}^{\text{op}} \rightarrow \mathbf{Cat}$. The link between indexed categories and logic was discovered by Lawvere in the 60ies [18, 19] ('quantifiers as adjoints'), and is at the heart of categorical logic.

Definition 4.19. Given indexed preorders $\mathcal{D}, \mathcal{E} : \mathbf{Set}^{\text{op}} \rightarrow \mathbf{Ord}$, an indexed monotone map $\sigma : \mathcal{D} \rightarrow \mathcal{E}$ is a family

$$\sigma_I : \mathcal{D}(I) \rightarrow \mathcal{E}(I) \quad (I \in \mathbf{Set})$$

of monotone functions, such that we have

$$(4.2) \quad \sigma_J(f^*(\varphi)) \cong f^*(\sigma_I(\varphi))$$

for all functions $f : J \rightarrow I$ and predicates $\varphi \in \mathcal{D}(I)$.

- Remarks 4.20.** (1) Indexed monotone maps are special cases of pseudo-natural transformations [17]. If we have equality in (4.2), we speak of a strict indexed monotone map, which is an instance of a 2-natural transformation.
- (2) Indexed preorders and indexed monotone maps form a category, which we denote by \mathbf{IOrd} . Composition of indexed monotone maps $\mathcal{C} \xrightarrow{\sigma} \mathcal{D} \xrightarrow{\tau} \mathcal{E}$ is defined by $(\tau \circ \sigma)_I(\varphi) = \tau_I(\sigma_I(\varphi))$ for $\varphi \in \mathcal{C}(I)$. The identity $\text{id}_{\mathcal{D}}$ of an indexed preorder \mathcal{D} is defined by $\text{id}_{\mathcal{D}, I}(\varphi) = \varphi$ for all $\varphi \in \mathcal{D}(I)$.

Definition 4.21. Let \mathcal{D}, \mathcal{E} be indexed preorders.

- (1) For indexed monotone maps $\sigma, \tau : \mathcal{D} \rightarrow \mathcal{E}$, we define

$$\sigma \leq \tau : \Leftrightarrow \forall I \in \mathbf{Set} . \sigma_I \leq \tau_I.$$

We say that σ and τ are isomorphic, and write $\sigma \cong \tau$, if $\sigma \leq \tau$ and $\tau \leq \sigma$.

- (2) An indexed monotone map $\sigma : \mathcal{D} \rightarrow \mathcal{E}$ is called an equivalence, if there exists a indexed monotone map $\tau : \mathcal{E} \rightarrow \mathcal{D}$ such that $\tau \circ \sigma \cong \text{id}_{\mathcal{D}}$, and $\sigma \circ \tau \cong \text{id}_{\mathcal{E}}$. In this case, τ is called an (indexed) weak inverse of σ .
- (3) We say that \mathcal{D} and \mathcal{E} are equivalent, and write $\mathcal{D} \simeq \mathcal{E}$, if there exists an equivalence $\sigma : \mathcal{D} \rightarrow \mathcal{E}$.

Lemma 4.22. An indexed monotone map $\sigma : \mathcal{D} \rightarrow \mathcal{E}$ is an equivalence, if and only if for every set I , the monotone map $\sigma_I : \mathcal{D}(I) \rightarrow \mathcal{E}(I)$ is order reflecting and essentially surjective.

Proof. By Remark 4.3, (1), every σ_I has a weak inverse $\tau_I : \mathcal{E}(I) \rightarrow \mathcal{D}(I)$. Together these τ_I give rise to an indexed weak inverse of σ . \square

4.4. Triposes. Next we consider a special kind of indexed Heyting preorders, called *triposes*, see [7].

Definition 4.23. A tripos is a functor $\mathcal{P} : \mathbf{Set}^{\text{op}} \rightarrow \mathbf{HPO}$ such that

- (1) For every function $f : J \rightarrow I$, the reindexing map $f^* : \mathcal{P}(I) \rightarrow \mathcal{P}(J)$ has a right adjoint $\forall_f : \mathcal{P}(J) \rightarrow \mathcal{P}(I)$.
- (2) If

$$(4.3) \quad \begin{array}{ccc} P & \xrightarrow{q} & K \\ p \downarrow & \lrcorner & \downarrow g \\ J & \xrightarrow{f} & I \end{array}$$

is a pullback square of sets and functions, then $\forall_q(p^*(\varphi)) \cong g^*(\forall_f(\varphi))$ for all $\varphi \in \mathcal{P}(J)$ (this is the Beck-Chevalley condition).

- (3) \mathcal{P} has a generic predicate, i.e. there exists a set \mathbf{Prop} , and a $\mathbf{tr} \in \mathcal{P}(\mathbf{Prop})$ such that for every set I and $\varphi \in \mathcal{P}(I)$ there exists a (not necessarily unique) function $\chi_\varphi : I \rightarrow \mathbf{Prop}$ with $\varphi \cong \chi_\varphi^*(\mathbf{tr})$.

Remark 4.24. (1) $\forall_f : \mathcal{P}(J) \rightarrow \mathcal{P}(I)$ is not required to preserve meets or implication.

- (2) The statement that the above square is a pullback, means explicitly that

$$\forall j \in J, k \in K. f(j) = g(k) \Leftrightarrow (\exists! x \in P. p(x) = j \wedge q(x) = k).$$

Lemma 4.25. Let \mathcal{D} and \mathcal{P} be indexed preorders, and assume that $\sigma : \mathcal{D} \rightarrow \mathcal{P}$ and $\tau : \mathcal{P} \rightarrow \mathcal{D}$ form an equivalence. If \mathcal{P} is a tripos, then so is \mathcal{D} .

Proof. This is because all the defining properties of a tripos are stable under equivalence, and can be transported along σ and τ . In particular:

- (1) for any set I , $\tau_I(\top)$ is a greatest element in $\mathcal{D}(I)$
- (2) meets in $\mathcal{D}(I)$ are given by $\varphi \wedge \psi = \tau_I(\sigma_I(\varphi) \wedge \sigma_I(\psi))$
- (3) Heyting implication in $\mathcal{D}(I)$ is given by $\varphi \rightarrow \psi = \tau_I(\sigma_I(\varphi) \rightarrow \sigma_I(\psi))$
- (4) universal quantification in \mathcal{D} can be defined by $\forall_f(\varphi) = \tau_I(\forall_f(\sigma_J(\varphi)))$ for $f : J \rightarrow I$ and $\varphi \in \mathcal{D}(J)$
- (5) a generic predicate for \mathcal{D} is given by $\tau_{\mathbf{Prop}}(\mathbf{tr})$ where $\mathbf{tr} \in \mathcal{P}(\mathbf{Prop})$ is the generic predicate of \mathcal{P}

□

5. CONSTRUCTING TRIPOSES FROM ORDERED STRUCTURES

In this section we show how to construct triposes –or weaker structures such as indexed meet-semilattices or indexed preorders– from ordered combinatory algebras or abstract Krivine structures. We also consider the relations between the different constructions.

5.1. From \mathcal{AKS} s to indexed preorders.

Definition 5.1. Let $\mathcal{K} = (\Lambda, \Pi, \dots)$ be an abstract Krivine structure, and let I be any set. The entailment relation \vdash in $\mathcal{P}(\Pi)^I$ is defined by

$$(5.1) \quad \varphi, \psi \in \mathcal{P}(\Pi)^I, \quad \varphi \vdash \psi \quad :\Leftrightarrow \quad \exists t \in \mathbf{QP} \quad \forall i \in I. t \perp \varphi(i) \Rightarrow \psi(i)$$

for $\varphi, \psi : I \rightarrow \mathcal{P}(\Pi)$. An element $t \in \Phi$ as above is said to be “a realizer of the entailment $\varphi \vdash \psi$ ”.

Remark 5.2. Notice that the entailment relation above, could have been defined using the arrow \Rightarrow_\perp , because $t \perp \varphi(i) \Rightarrow \psi(i)$ if and only if $t \perp \varphi(i) \Rightarrow_\perp \psi(i)$, compare with Remark 4.10.

Lemma 5.3. Let \mathcal{K} be an abstract Krivine structure.

- (1) For any set I , the entailment relation \vdash is a preorder on $\mathcal{P}(\Pi)^I$.
- (2) For any function $f : J \rightarrow I$, precomposition defines a monotone map

$$f^* : (\mathcal{P}(\Pi)^I, \vdash) \rightarrow (\mathcal{P}(\Pi)^J, \vdash), \quad \varphi \mapsto \varphi \circ f.$$

- (3) The preceding constructions give an indexed preorder

$$\mathcal{P}(\mathcal{K}) : \mathbf{Set}^{\mathbf{op}} \rightarrow \mathbf{Ord}, \quad I \mapsto (\mathcal{P}(\Pi)^I, \vdash), \quad f \mapsto f^*.$$

Proof. Ad 1 – This is proved in the same way as Lemma 4.11.

Ad 2 – Let $\varphi, \psi : I \rightarrow \mathcal{P}(\Pi)$. If $t \in \Phi$ is a realizer of $\varphi \vdash \psi$, then it is also a realizer of $\varphi \circ f \vdash \psi \circ f$, thus f^* is monotone.

Ad 3 – We check the functoriality condition, i.e. $g^* \circ f^* = (f \circ g)^*$ and $\text{id}_I^* = \text{id}_{A^I}$ for $K \xrightarrow{g} J \xrightarrow{f} I$. This follows from associativity and unit laws for composition. □

Definition 5.4. The indexed preorder $\mathcal{P}_\perp(\mathcal{K}) : \mathbf{Set}^{\text{op}} \rightarrow \mathbf{Ord}$ is defined by

$$\mathcal{P}_\perp(\mathcal{K})(I) = (\mathcal{P}_\perp(\Pi)^I, \vdash), \quad f \mapsto f^*$$

where the order on $\mathcal{P}_\perp(\Pi)^I$ is the restriction of the entailment order on $\mathcal{P}(\Pi)^I$ to predicates with values in $\mathcal{P}_\perp(\Pi)$.

Lemma 5.5. The canonical inclusion $\mathcal{P}_\perp(\mathcal{K}) \hookrightarrow \mathcal{P}(\mathcal{K})$ is an equivalence of indexed preorders.

Proof. By Lemma 4.22 it suffices to show that the inclusion

$$(\mathcal{P}_\perp(\Pi)^I, \vdash) \hookrightarrow (\mathcal{P}(\Pi)^I, \vdash)$$

is an equivalence for all sets I and this is proved in the same way as Lemma 4.12. \square

5.2. From OCA s to indexed meet-semilattices.

Definition 5.6. Let (\mathcal{A}, Φ) be a filtered OCA . The entailment relation $\vdash \subseteq A^I \times A^I$ is defined by

$$(5.2) \quad \varphi \vdash \psi \quad :\Leftrightarrow \quad \exists r \in \Phi \, \forall i \in I. r(\varphi(i)) \leq \psi(i)$$

for $\varphi, \psi : I \rightarrow A$. An $r \in \Phi$ as above is said to be “a realizer of the entailment $\varphi \vdash \psi$ ”.

Lemma 5.7. Let (\mathcal{A}, Φ) be a filtered OCA .

- (1) For any set I , the entailment relation \vdash is a preorder on A^I , and (A^I, \vdash) is a meet-semi-lattice with the following definitions: $\top : I \rightarrow A$; $\top(i) = \top = k$ and $\varphi \wedge \psi$ of two functions $\varphi, \psi : I \rightarrow A$ is $(\varphi \wedge \psi)(i) = \varphi(i) \wedge \psi(i)$.
- (2) For any function $f : J \rightarrow I$, precomposition with f defines a meet preserving monotone map

$$f^* : (A^I, \vdash) \rightarrow (A^J, \vdash), \quad \varphi \mapsto \varphi \circ f.$$

- (3) The preceding constructions define an indexed meet-semi-lattice

$$\mathcal{P}(\mathcal{A}) : \mathbf{Set}^{\text{op}} \rightarrow \mathbf{SLat}, \quad I \mapsto (A^I, \vdash), \quad f \mapsto f^*.$$

Proof. Ad 1 – We use in the proof of this assertion the realizers exhibited in Lemma 4.14.

Ad 2 – Let $\varphi, \psi : I \rightarrow A$. If $r \in \Phi$ is a realizer of $\varphi \vdash \psi$, then it is also a realizer of $\varphi \circ f \vdash \psi \circ f$, thus f^* is monotone. For meets, we have

$$((\varphi \wedge \psi) \circ f)(j) = p(\varphi(fj)\psi(fj)) = ((\varphi \circ f) \wedge (\psi \circ f))(j)$$

for all $j \in J$, which means $f^*(\varphi \wedge \psi) = f^*(\varphi) \wedge f^*(\psi)$. Preservation of \top is shown in the same way.

Ad 3 – It remains to check functoriality, i.e. $g^* \circ f^* = (f \circ g)^*$ and $\text{id}_I^* = \text{id}_{A^I}$ for $K \xrightarrow{g} J \xrightarrow{f} I$. This follows from associativity and unit laws for composition. \square

5.3. From $\mathcal{I}\text{OCA}$ s to triposes.

Next we show that if the OCA considered above has the necessary additional structure to make it a $\mathcal{I}\text{OCA}$, the indexed meet semi-lattice just constructed is in fact a tripos.

For any $\mathcal{I}\text{OCA}$, $\mathcal{A} = (A, \leq, \text{app}, \text{imp}, \Phi, k, s, e)$, the quintuple $(A, \leq, \text{app}, k, s)$ is an OCA –that we also call \mathcal{A} –, and Φ is a filter on it. Thus, we can construct the indexed meet-semi-lattice $\mathcal{P}(\mathcal{A})$ from Definition 5.6.

Theorem 5.8. If $\mathcal{A} = (A, \leq, \text{app}, \text{imp}, \Phi, k, s, e)$ is a $\mathcal{I}\text{OCA}$, then $\mathcal{P}(\mathcal{A})$ is a tripos. Moreover, if the $\mathcal{I}\text{OCA}$ is a κOCA –with combinator $c \in \Phi$ – then $\neg\neg\varphi \vdash \varphi$, with $\neg\varphi := \varphi \rightarrow \perp$.

Proof. We know that $\mathcal{P}(\mathcal{A})$ is an indexed meet-semi-lattice, and it remains to show that it has implication, universal quantification, and a generic predicate.

For $\psi, \theta : I \rightarrow A$, we define $\varphi \rightarrow \psi$ by

$$(\psi \rightarrow \theta)(i) = \varphi(i) \rightarrow \psi(i)$$

To see that this gives a Heyting implication on (A^I, \vdash) , we have to check that

$$\varphi \vdash \psi \rightarrow \theta \quad \Leftrightarrow \quad \varphi \wedge \psi \vdash \theta$$

where $(\varphi \wedge \psi)(i) = \mathbf{p}\varphi(i)\psi(i)$. In a similar manner as before, the assertion is shown in the same way as Theorem 4.15.

Universal quantification of a predicate $\psi : J \rightarrow A$ along a function $f : J \rightarrow I$ is defined by

$$\forall_f(\psi)(i) = \inf_{f(j)=i} \psi(j)$$

With this definition it follows directly that for any $\varphi : I \rightarrow A$ and $r \in \Phi$ we have

$$\begin{aligned} \forall j \in J. r \varphi(f(j)) &\leq \psi(j) \\ \Leftrightarrow \forall i \in I. r \varphi(i) &\leq \forall_f(\psi)(i), \end{aligned}$$

which means that $f^*\varphi \vdash \psi$ if and only if $\varphi \vdash \forall_f\psi$ (with the same realizer), hence Remark 4.3, (2) implies that \forall_f is right adjoint to f^* .

For the Beck-Chevalley condition, consider the pullback square (4.3) in Definition 4.23, and let $\varphi : J \rightarrow I$. For $k \in K$ we have

$$\begin{aligned} g^*(\forall_f(\varphi))(k) &= \inf_{fj=gk} \varphi(j) \\ \text{and } \forall_q(p^*(\varphi))(k) &= \inf_{\substack{x \in P \\ qx=k}} \varphi(p(x)). \end{aligned}$$

In the first case, the infimum is taken over the set $\{j \in J \mid f(j) = g(k)\}$, and in the second case over the set $\{j \in J \mid \exists x \in P. p(x) = j, q(x) = k\}$. These two sets are equal since the square is a pullback (thus the Beck Chevalley condition holds even up to equality).

Finally, a generic predicate for $\mathcal{P}(\mathcal{A})$ is given by $\text{id}_A \in \mathcal{P}(\mathcal{A})(A)$.

The fact that $\neg\neg\varphi \vdash \varphi$ for all predicates φ , follows directly from Lemma 4.16.

□

5.4. From \mathcal{AKS} s to \mathcal{KOCA} s.

Next, we recall the construction due to Streicher (see [23]) that starting from an \mathcal{AKS} abbreviated as \mathcal{K} produces a \mathcal{KOCA} —that we call $\mathcal{A}_{\mathcal{K}}$ —and show that they induce isomorphic indexed preorders—in fact triposes—.

Definition 5.9. *Given an \mathcal{AKS} :*

$$\mathcal{K} = (\Lambda, \Pi, \perp, \text{push}, \text{app}, \text{store}, \kappa, \mathbf{s}, \text{cc}, \text{QP})$$

define

$$\mathcal{A}_{\mathcal{K}} = (A, \leq, \text{app}, \text{imp}, \mathbf{k}, \mathbf{s}, \mathbf{c}, \mathbf{e}, \Phi)$$

as follows.

- (1) $(A, \leq) = (\mathcal{P}_{\perp}(\Pi), \supseteq)$;
- (2) $\text{app}(P, Q) = P \circ_{\perp} Q = (\perp(\perp Q \rightsquigarrow P))^{\perp}$, $\text{imp}(P, Q) = P \Rightarrow_{\perp} Q = (\perp(\perp P \cdot Q))^{\perp}$;
- (3) $\mathbf{k} = \{\kappa\}^{\perp}$, $\mathbf{s} = \{\mathbf{s}\}^{\perp}$, $\mathbf{c} = \{\text{cc}\}^{\perp}$, $\mathbf{e} = \{\mathbf{E}\mathbf{E}\}^{\perp}$, where $\mathbf{E} = \mathbf{s}(\kappa(\mathbf{s}\kappa\kappa))$;
- (4) $\Phi = \{P \in \mathcal{P}_{\perp}(\Pi) \mid \exists t \in \text{QP}. t \perp P\}$.

If $a, b \in A$ we write $ab := \text{app}(a, b)$ and $a \rightarrow b := \text{imp}(a, b)$. See Definitions 2.5, 2.7 and 2.8.

We recall the following important theorem from [23] and write down a short proof for later use.

Theorem 5.10. *Let \mathcal{K} be an \mathcal{AKS} and consider the structure $\mathcal{A}_{\mathcal{K}}$ presented in Definition 5.9.*

- (1) *Then, $\mathcal{A}_{\mathcal{K}}$ is a \mathcal{KOCA} .*

(2) The associated indexed preorders $\mathcal{P}_\perp(\mathcal{K})$ and $\mathcal{P}(\mathcal{A}_\mathcal{K})$ are isomorphic.

Proof. (1) The order is clearly inf complete as we observed in Remark 2.2. The fact that the implication and application satisfy the monotonicity properties, is clear. The implication \rightarrow satisfies the *half adjunction property*: if $a \leq (b \rightarrow c)$ then $ab \leq c$ as was established in Theorem 2.6.

Next we prove that $kab \leq a$. Lemma 2.10 (2) guarantees that for all $a, b \in A$, $\kappa \in {}^\perp({}^\perp a.({}^\perp b.a))$. This assertion means that $\{\kappa\} \subseteq {}^\perp({}^\perp a.({}^\perp b.a))$ and then $k \supseteq \left({}^\perp({}^\perp a.({}^\perp b.a))\right)^\perp \supseteq {}^\perp a.({}^\perp b.a)$ that can be written as ${}^\perp a \rightsquigarrow k \supseteq {}^\perp b.a$. Moreover, from Definition 2.5, (2.1) we deduce that $k \circ_\perp a \supseteq {}^\perp a \rightsquigarrow k \supseteq {}^\perp b.a$, i.e. $k \circ_\perp a \leq (b \rightarrow a)$ –compare with Definition 2.5–. Using the half adjunction property (Theorem 2.6), we deduce that $kab \leq a$.

The condition $sabc \leq (ac)(bc)$ can be proved as follows. Take $t \perp a$, $s \perp b$, $u \perp c$ then using Lemma 2.9 we deduce that $(su) \perp bc$ and $(tu) \perp ac$ and also that $(tu)(su) \perp (ac)(bc)$ and if $\pi \in (ac)(bc)$ is an arbitrary element we conclude that $(tu)(su) \perp \pi$. Then by the Definition 2.7, (S3) we conclude that $s \perp t.s.u.\pi$. Hence we have proved that $s \perp {}^\perp a.{}^\perp b.{}^\perp c.(ac)(bc)$ or $s \in {}^\perp({}^\perp a.{}^\perp b.{}^\perp c.(ac)(bc))$ or equivalently that $s \supseteq {}^\perp a.{}^\perp b.{}^\perp c.(ac)(bc)$.

Assume now that we have a situation as follows: $x, y \in A$, $z \subseteq \Pi$ with ${}^\perp x.z \subseteq y$, clearly it follows from Remark 2.4 that $z \subseteq {}^\perp x \rightsquigarrow y$

If we apply repeatedly the above observation to ${}^\perp a.{}^\perp b.{}^\perp c.(ac)(bc) \subseteq s$ we deduce that $(ac)(bc) \subseteq sabc$, and the proof of this part is finished.

The proof that e as introduced in Definition 5.9, is an adjunction is the content of Theorem 2.13.

The proof that $\Phi \subseteq A$ is a filter that contains k, s, e is the following. The subset Φ is closed under application because if $f, g \in \Phi$, i.e. if we have $t_f \in {}^\perp f \cap \text{QP}$ and $t_g \in {}^\perp g \cap \text{QP}$ then $t_f t_g \in {}^\perp f {}^\perp g \cap \text{QP} \subseteq {}^\perp (f \circ_\perp g) \cap \text{QP}$ (Lemma 2.9). Moreover, $k, s, e \in \Phi$ because $\kappa \in {}^\perp k \cap \text{QP}$, $s \in {}^\perp s \cap \text{QP}$ and $e \in {}^\perp e \cap \text{QP}$.

Finally, as we took $c = \{cc\}^\perp$, it is clear that: $cc \in {}^\perp c \cap \text{QP}$. Moreover, we proved in Lemma 2.10 that $cc \in {}^\perp(((a \rightarrow b) \rightarrow a) \rightarrow a)$, that implies that $c \supseteq ({}^\perp(((a \rightarrow b) \rightarrow a) \rightarrow a))^\perp = (((a \rightarrow b) \rightarrow a) \rightarrow a)$, i.e. $c \leq (((a \rightarrow b) \rightarrow a) \rightarrow a)$.

(2) In both cases the predicates on a set I are functions $\varphi, \psi : I \rightarrow \mathcal{P}_\perp(\Pi)$, so we only have to check that the two definitions of entailment coincide. The entailment in $\mathcal{P}(\mathcal{A}_\mathcal{K})$ is given by

$$\exists P \in \Phi \forall i \in I. P\varphi(i) \leq \psi(i)$$

which using the adjunction and substituting P by eP can be formulated equivalently as:

$$\exists P \in \Phi \forall i \in I. P \leq \varphi(i) \rightarrow \psi(i)$$

As to the equivalence we have that:

$$\begin{aligned} & \exists P \in \Phi \forall i \in I. P \leq \varphi(i) \rightarrow \psi(i) \\ \Leftrightarrow & \exists t \in \text{QP} \forall i \in I. \{t\}^\perp \supseteq ({}^\perp({}^\perp \varphi(i) \cdot \psi(i)))^\perp \\ \Leftrightarrow & \exists t \in \text{QP} \forall i \in I. t \perp ({}^\perp({}^\perp \varphi(i) \cdot \psi(i)))^\perp \\ \Leftrightarrow & \exists t \in \text{QP} \forall i \in I. t \perp {}^\perp \varphi(i) \cdot \psi(i) \\ \Leftrightarrow & \exists t \in \text{QP} \forall i \in I. t \perp \varphi(i) \Rightarrow \psi(i) \end{aligned}$$

and the last line is the definition of entailment in $\mathcal{P}_\perp(\mathcal{K})$.

□

5.5. From ${}^{\mathcal{K}}\mathcal{OCA}$ s to \mathcal{AKS} s.

In order to complete our program to set up the foundations of realizability in terms of ${}^{\mathcal{K}}\mathcal{OCA}$ s, we reverse the construction presented in Subsection 5.4 and show how to construct from an ${}^{\mathcal{K}}\mathcal{OCA}$ called \mathcal{A} , an \mathcal{AKS} named as $\mathcal{K}_{\mathcal{A}}$. Then, we prove that the corresponding triposes are equivalent.

Definition 5.11. *Given a ${}^{\mathcal{K}}\mathcal{OCA}$*

$$\mathcal{A} = (A, \leq, \text{app}_{\mathcal{A}}, \text{imp}, \mathbf{k}, \mathbf{s}, \mathbf{c}, \mathbf{e}, \Phi)$$

we define the structure:

$$\mathcal{K}_{\mathcal{A}} = (\Lambda, \Pi, \perp, \text{push}, \text{app}, \text{store}, \mathbf{k}, \mathbf{s}, \mathbf{cc}, \text{QP})$$

as follows.

- (1) $\Lambda = \Pi := A$
- (2) $\perp := \leq$, *i.e.* $s \perp \pi :\Leftrightarrow s \leq \pi$
- (3) $\text{push}(s, \pi) := \text{imp}(s, \pi) = s \rightarrow \pi$, $\text{app}(s, t) := \text{app}_{\mathcal{A}}(s, t) = st$, $\text{store}(\pi) := \neg\pi$
- (4) $\mathbf{k} := \mathbf{e}(\mathbf{bek})$, $\mathbf{s} := \mathbf{e}(\mathbf{b}(\mathbf{be}(\mathbf{be}))\mathbf{s})$, $\mathbf{cc} := \mathbf{ec}$
- (5) $\text{QP} := \Phi$

Here, \mathbf{b} is an abbreviation for $\mathbf{s}(\mathbf{k}\mathbf{s})\mathbf{k}$, which has the property that $\mathbf{b}abc \leq a(\mathbf{b}c)$ for all $a, b, c \in A$, and $\neg\pi$ is a shorthand for $\pi \rightarrow \perp$ and $\perp := \inf(A)$.

Theorem 5.12. *In the notations of Definition 5.11 the structure $\mathcal{K}_{\mathcal{A}}$ is an \mathcal{AKS} .*

Proof. It is clear that QP is closed under application and contains $\mathbf{k}, \mathbf{s}, \mathbf{cc}$, and it remains to check the axioms about the orthogonality relation (see Definition 2.7). Substituting the above definitions, these axioms become:

- (S1) $t \leq u \rightarrow \pi \Rightarrow tu \leq \pi$
- (S2) $t \leq \pi \Rightarrow \mathbf{e}(\mathbf{bek}) \leq t \rightarrow u \rightarrow \pi$
- (S3) $tv(uv) \leq \pi \Rightarrow \mathbf{e}(\mathbf{b}(\mathbf{be}(\mathbf{be}))\mathbf{s}) \leq t \rightarrow u \rightarrow v \rightarrow \pi$
- (S4) $t \leq \neg\pi \rightarrow \pi \Rightarrow \mathbf{ec} \leq t \rightarrow \pi$
- (S5) $t \leq \pi \Rightarrow \neg\pi \leq t \rightarrow \pi', \quad \forall \pi'$

(S1) follows from Definition 3.8, (PA), and (S5) follows from monotonicity of the arrow in its second argument and the antitonicity in the first.

(S2) is shown by the following derivation.

$$\frac{\frac{\frac{t \leq \pi}{ktu \leq \pi}}{\mathbf{e}(kt) \leq u \rightarrow \pi}}{\mathbf{bekt} \leq u \rightarrow \pi}}{\mathbf{e}(\mathbf{bek}) \leq t \rightarrow u \rightarrow \pi}$$

(S3) is proved using repeatedly the basic properties of \mathbf{b} and \mathbf{e} as follows:

$$\frac{\frac{\frac{tv(uv) \leq \pi}{stuv \leq \pi}}{\mathbf{e}(stu) \leq v \rightarrow \pi}}{\mathbf{be}(st)u \leq v \rightarrow \pi}}{\mathbf{e}(\mathbf{be}(st)) \leq u \rightarrow v \rightarrow \pi}}{\mathbf{be}(\mathbf{be})(st) \leq u \rightarrow v \rightarrow \pi}}{\mathbf{b}(\mathbf{be}(\mathbf{be}))st \leq u \rightarrow v \rightarrow \pi}}{\mathbf{e}(\mathbf{b}(\mathbf{be}(\mathbf{be}))\mathbf{s}) \leq t \rightarrow u \rightarrow v \rightarrow \pi}$$

Finally, (S4) is proved using the basic property of \mathbf{c} – Definition 3.8, (PC), the monotony of the application and the definition of \mathbf{e} – as follows:

$$\begin{array}{c}
 \text{(PC)} \\
 \hline
 \frac{\mathbf{c} \leq (\neg\pi \rightarrow \pi) \rightarrow \pi \quad t \leq \neg\pi \rightarrow \pi}{ct \leq ((\neg\pi \rightarrow \pi) \rightarrow \pi)(\neg\pi \rightarrow \pi)} \quad \frac{(\neg\pi \rightarrow \pi) \rightarrow \pi \leq (\neg\pi \rightarrow \pi) \rightarrow \pi}{((\neg\pi \rightarrow \pi) \rightarrow \pi)(\neg\pi \rightarrow \pi) \leq \pi} \\
 \hline
 \frac{ct \leq \pi}{\mathbf{e}\mathbf{c} \leq t \rightarrow \pi}
 \end{array}$$

□

Definition 5.13. Let (D, \leq) be a preorder.

- (1) A principal filter in D is a subset of D of the form

$$\uparrow d_0 := \{d \in D \mid d_0 \leq d\}.$$

for some $d_0 \in D$.

- (2) Dually, a principal ideal in D is a subset of the form

$$\downarrow d_0 := \{d \in D \mid d \leq d_0\}.$$

for $d_0 \in D$.

Lemma 5.14. Let \mathcal{A} be a κOCA structure, and $\mathcal{K}_{\mathcal{A}}$ the \mathcal{AKS} induced via the construction in Definition 5.11.

- (1) For $U \subseteq A$ we have ${}^\perp U = \downarrow(\inf U)$, and $U^\perp = \uparrow(\sup U)$.
(2) For $a \in A$ we have $\inf(\uparrow a) = a = \sup(\downarrow a)$
(3) The set $\mathcal{P}_\perp(\Pi)$ consists precisely of the principal filters in A , and the maps

$$f : A \rightarrow \mathcal{P}_\perp(\Pi), a \mapsto \uparrow a \quad \text{and} \quad g : \mathcal{P}_\perp(\Pi) \rightarrow A, P \mapsto \inf P,$$

are mutually inverse and establish a bijection between A and $\mathcal{P}_\perp(\Pi)$.

- (4) For $P, Q \in \mathcal{P}_\perp(\Pi)$ we have $\inf(P \Rightarrow_\perp Q) = \inf(P \Rightarrow Q) = \inf P \rightarrow \inf Q$.

Proof. Ad 1 – ${}^\perp U$ is the set of lower bounds of U , and $\inf U$ is the greatest lower bound. An element $a \in A$ is a lower bound of U if and only if it is smaller than the greatest lower bound. The second claim is just the dual (recall that this duality is valid in a lattice).

Ad 2 – a is a lower bound of $\uparrow a$, and since $a \in \uparrow a$ any other lower bound must be smaller. Thus a is the greatest lower bound. The second part is symmetric.

Ad 3 – For $P \subseteq A$ we have $({}^\perp P)^\perp = (\downarrow(\inf P))^\perp = \uparrow(\inf P)$, thus all $({}^\perp(-))^\perp$ -stable sets are principal filters.

Conversely, for a principal filter of the form $\uparrow a$ and using the previous parts of this Lemma, we have that $({}^\perp \uparrow a)^\perp = (\downarrow(\inf(\uparrow a)))^\perp = \uparrow(\sup(\downarrow(\inf(\uparrow a)))) = \uparrow(\sup(\downarrow a)) = \uparrow a$.

To see that f and g are mutually inverse, take first $a \in A$. Then $g(f(a)) = \inf(\uparrow a) = a$. In the other direction, let $P \in \mathcal{P}_\perp(\Pi)$. We know that P is a principal filter, thus $P = \uparrow a$ for some $a \in A$ and we have $f(g(P)) = \uparrow(\inf P) = \uparrow(\inf(\uparrow a)) = \uparrow a = P$.

Ad 4 – The fact that $\inf(P \Rightarrow_\perp Q) = \inf(P \Rightarrow Q)$ follows also from the previous results. Indeed, we have that $\inf(P \Rightarrow_\perp Q) = \inf({}^\perp(P \Rightarrow Q))^\perp = \inf((\downarrow(\inf(P \Rightarrow Q)))^\perp) = \inf(\downarrow(\inf P \rightarrow \inf Q)^\perp) = \inf((\downarrow(a \rightarrow b))^\perp) = \inf(\uparrow(\sup(\downarrow(a \rightarrow b)))) = a \rightarrow b = \inf P \rightarrow \inf Q = \inf(P \Rightarrow Q)$. In the above computations we used that: $P = \uparrow a$, $Q = \uparrow b$ and the parts (1), (2) and (3) already proved. The last equality is proved below.

From the preceding claim we know that given P, Q as above, there are elements $a, b \in A$ such that $P = \uparrow a$ and $Q = \uparrow b$. We have

$$\uparrow a \Rightarrow \uparrow b = {}^\perp(\uparrow a) \cdot \uparrow b = \downarrow(\inf(\uparrow a)) \cdot \uparrow b = \downarrow a \cdot \uparrow b = \{c \rightarrow d \mid c \leq a, b \leq d\}$$

and thus $\inf(\uparrow a \Rightarrow \uparrow b) = a \rightarrow b$ by monotonicity of the arrow. □

Theorem 5.15. The associated indexed triposes $\mathcal{P}(\mathcal{A})$ and $\mathcal{P}_\perp(\mathcal{K}_{\mathcal{A}})$ are equivalent (see Definitions 5.7-3 and in 5.3-3 respectively).

Proof. Let I be a set. The elements of $\mathcal{P}(\mathcal{A})(I)$ are functions $\varphi : I \rightarrow A$, and the elements of $\mathcal{P}_\perp(\mathcal{K}_\mathcal{A})(I)$ are functions $\widehat{\varphi} : I \rightarrow \mathcal{P}_\perp(\Pi)$.

Post-composition with f and g from Lemma 5.14-3 induces a bijection between $\mathcal{P}(\mathcal{A})(I)$ and $\mathcal{P}_\perp(\mathcal{K}_\mathcal{A})(I)$, and it remains to show that this bijection is compatible with the entailment orderings.

Let $\varphi, \psi : I \rightarrow A$ be two predicates in $\mathcal{P}(\mathcal{A})(I)$, with corresponding predicates $f \circ \varphi, f \circ \psi$ in $\mathcal{P}_\perp(\mathcal{K}_\mathcal{A})(I)$. Then we can reformulate the entailment $f \circ \varphi \vdash f \circ \psi$ in $\mathcal{P}_\perp(\mathcal{K}_\mathcal{A})(I)$ as follows:

$$\begin{aligned} f \circ \varphi \vdash f \circ \psi &\Leftrightarrow \exists a \in \Phi \forall i \in I. a \perp \uparrow(\varphi i) \Rightarrow \uparrow(\psi i) \\ &\Leftrightarrow \exists a \in \Phi \forall i \in I \forall b \in [\uparrow(\varphi i) \Rightarrow \uparrow(\psi i)] \quad \text{then} \quad a \leq b \\ &\Leftrightarrow \exists a \in \Phi \forall i \in I. a \leq \inf[\uparrow(\varphi i) \Rightarrow \uparrow(\psi i)] \\ &\Leftrightarrow \exists a \in \Phi \forall i \in I. a \leq \varphi(i) \rightarrow \psi(i), \end{aligned}$$

and this is equivalent to the entailment $\varphi \vdash \psi$ in $\mathcal{P}(\mathcal{A})(I)$:

$$\exists a \in \Phi \forall i \in I. a\varphi(i) \leq \psi(i)$$

by axioms (PA) and (PE) in Definition 3.8. □

6. INTERNAL REALIZABILITY IN $\mathcal{K}\mathcal{OCA}$ S

We have shown that the class of ordered combinatory algebras that, besides a filter of distinguished truth values are equipped with an implication, an adjunction and satisfy a completeness condition with respect to the infimum over arbitrary subsets – i.e.: $\mathcal{K}\mathcal{OCA}$ s – is rich enough as to allow the Tripes construction and as such its objects can be taken as the basis of the categorical perspective on classical realizability –à la Streicher–. In this section we show that we can define realizability in this type of combinatory algebras, and thus, to define realizability in higher-order arithmetic.

Definition 6.1. Consider a set of constants of kinds, one of its elements is denoted by o . The language of kinds is given by the following grammar:

$$\sigma, \tau ::= c \quad | \quad \sigma \rightarrow \tau$$

Consider an infinite set of variables labeled by kinds x^τ . Suppose that we have infinitely many variables labeled of the kind τ for each kind τ . Consider also a set of constants a^τ, b^σ, \dots labeled with a kind. The language \mathcal{L}^ω of order ω is defined by the following grammar:

$$M^\sigma, N^{\sigma \rightarrow \tau}, A^o, B^o ::= x^\sigma \quad | \quad a^\sigma \quad | \quad (\lambda x^\sigma. M^\tau)^{\sigma \rightarrow \tau} \quad | \quad (N^{\sigma \rightarrow \tau} M^\sigma)^\tau \quad | \quad (A^o \Rightarrow B^o)^o \quad | \quad (\forall x^\tau. A^o)^o$$

o represents the type of truth values. The expressions labeled by o are called “formulae”. The symbols \rightarrow and \Rightarrow , when iterated, are associated on the right side. On the other hand, the application, when iterated, are associated on the left side.

Definition 6.2. Let A be a $\mathcal{K}\mathcal{OCA}$ and consider a set of variables $\mathcal{V} = \{x_1, x_2, \dots\}$. A declaration is a string of the shape $x_i : A^o$. A context is a string of the shape $x_1 : A_1^o, \dots, x_k : A_k^o$, i.e.: contexts are finite sequences of declarations. The contexts will be often denoted by capital Greek letters: Δ, Γ, Σ . A sequent is a string of the shape $x_1 : A_1^o, \dots, x_k : A_k^o \vdash p : B^o$ where p is a polynomial of $A[x_1, \dots, x_k]$. The left side of a sequent is a context. When we do not make the declarations of the context of a sequent explicit, we will write it as $\Gamma \vdash p : B^o$. Typing rules are trees with leaves of the shape

$$\frac{S_1 \quad \dots \quad S_h}{S_{h+1}} \text{ (Rule)}$$

where $h \geq 0$ and S_1, \dots, S_{h+1} are sequents. The typing rules for \mathcal{L}^ω are the following:

$$(where \ x_i : A_i^o \text{ appears in } \Gamma) \frac{}{\Gamma \vdash x_i : A_i^o} (ax)$$

$$\frac{\Gamma, x : A^o \vdash p : B^o}{\Gamma \vdash e(\lambda^* x p) : (A^o \Rightarrow B^o)^o} (\rightarrow_i)$$

$$\frac{\Gamma \vdash p : (A^o \Rightarrow B^o)^o \quad \Gamma \vdash q : A^o}{\Gamma \vdash pq : B^o} (\rightarrow_e)$$

$$(where\ x^\sigma\ does\ not\ appear\ free\ in\ \Gamma) \frac{\Gamma \vdash p : A^o}{\Gamma \vdash p : (\forall x^\sigma A^o)^o} (\forall_i)$$

$$\frac{\Gamma \vdash p : (\forall x^\sigma A^o)^o}{\Gamma \vdash p : (A^o\{x^\sigma := M^\sigma\})^o} (\forall_e)$$

Definition 6.3. Let us consider $\mathcal{A} = (A, \leq, \text{app}, \rightarrow, \Phi, \mathbf{k}, \mathbf{s}, \mathbf{e}, \mathbf{c})$ a κOCA . We define the interpretation of \mathcal{L}^ω as follows:

- (1) For kinds: The interpretation of a constant c is a set $\llbracket c \rrbracket$. In particular, the constant o is interpreted as the underlying set of \mathcal{A} , i.e.: $\llbracket o \rrbracket = A$. Given two kinds σ, τ , the interpretation $\llbracket \sigma \rightarrow \tau \rrbracket$ is the set of functions $\llbracket \tau \rrbracket^{\llbracket \sigma \rrbracket}$.
- (2) For expressions: In order to interpret expressions, we start choosing an assignment \mathbf{a} for the variables x^σ such that $\mathbf{a}(x^\sigma) \in \llbracket \sigma \rrbracket$. As it is usual in semantics, the substitution-like notation $\{x^\sigma := s\}$ affecting an assignment \mathbf{a} –or an interpretation using $\mathbf{a}-$, modifies it by redefining \mathbf{a} over x^σ as the statement $\mathbf{a}\{x^\sigma := s\}(x^\sigma) := s$. We proceed similarly for interpretations.
 - For an expression of the shape x^σ , its interpretation is $\llbracket x^\sigma \rrbracket = \mathbf{a}(x^\sigma)$.
 - For an expression of the shape $\lambda x^\sigma M^\tau$, its interpretation is the function $\llbracket \lambda x^\sigma M^\tau \rrbracket \in \llbracket \sigma \rightarrow \tau \rrbracket$ defined as $\llbracket \lambda x^\sigma M^\tau \rrbracket(s) := \llbracket M^\tau \rrbracket\{x^\sigma := s\}$ for all $s \in \llbracket \sigma \rrbracket$.
 - For an expression of the shape $(N^{\sigma \rightarrow \tau} M^\sigma)^\tau$ its interpretation is $\llbracket (N^{\sigma \rightarrow \tau} M^\sigma)^\tau \rrbracket := \llbracket N^{\sigma \rightarrow \tau} \rrbracket(\llbracket M^\sigma \rrbracket)$.
 - For an expression of the shape $(A^o \Rightarrow B^o)^o$ its interpretation is $\llbracket (A^o \Rightarrow B^o)^o \rrbracket := \llbracket A^o \rrbracket \rightarrow \llbracket B^o \rrbracket$.
 - For an expression of the shape $(\forall x^\sigma A^o)^o$ its interpretation is

$$\llbracket (\forall x^\sigma A^o)^o \rrbracket := \inf \{ \llbracket A^o \rrbracket\{x^\sigma := s\} \mid s \in \llbracket \sigma \rrbracket \}$$

We say that \mathcal{A} satisfies a sequent $x_1 : A_1^o, \dots, x_k : A_k^o \vdash p : B^o$ if and only if for all assignment \mathbf{a} and for all $b_1, \dots, b_k \in A$, if $b_1 \leq \llbracket A_1^o \rrbracket, \dots, b_k \leq \llbracket A_k^o \rrbracket$ then $p\{x_1 := b_1, \dots, x_k := b_k\} \leq \llbracket B^o \rrbracket$. In this case we write that: $\mathcal{A} \models x_1 : A_1^o, \dots, x_k : A_k^o \vdash p : B^o$.

A rule:

$$\frac{S_1 \quad \dots \quad S_h}{S_{h+1}} (\text{Rule})$$

is said to be adequate if and only if for every $\mathcal{A} \in \kappa\text{OCA}$, if $\mathcal{A} \models S_1, \dots, S_h$ then $\mathcal{A} \models S_{h+1}$.

Theorem 6.4. The rules of the typing system appearing in Definition 6.2, are adequate.

Proof. For (ax) is evident.

For the implication rules:

- $(\rightarrow)_i$ Assume $\mathcal{A} \models \Gamma, x : A^o \vdash p : B^o$ where $\Gamma = x_1 : A_1^o, \dots, x_k : A_k^o$. Consider an assignment \mathbf{a} and $b_1, \dots, b_k \in A$ such that $b_i \leq \llbracket A_i^o \rrbracket$. We get:

$$\begin{aligned} (\lambda^* x p)\{x_1 := b_1, \dots, x_k := b_k\} \llbracket A^o \rrbracket &= (\lambda^* x p\{x_1 := b_1, \dots, x_k := b_k\}) \llbracket A^o \rrbracket \leq \\ & p\{x_1 := b_1, \dots, x_k := b_k, x := \llbracket A^o \rrbracket\} \leq \\ & \llbracket B^o \rrbracket \end{aligned}$$

the last inequality by the assumption $\mathcal{A} \models \Gamma, x : A^o \vdash p : B^o$.

Applying the adjunction property we deduce that $\mathbf{e}(\lambda^* x p)\{x_1 := b_1, \dots, x_k := b_k\} \leq \llbracket (A^o \Rightarrow B^o)^o \rrbracket$. Since the above is valid for all the assignments, we conclude $\mathcal{A} \models \Gamma \vdash \mathbf{e}(\lambda^* x p) : (A^o \Rightarrow B^o)^o$.

- $(\rightarrow)_e$ Assume $\mathcal{A} \models \Gamma \vdash p : (A^o \Rightarrow B^o)^o$ and $\mathcal{A} \models \Gamma \vdash q : A^o$ where $\Gamma = x_1 : A_1^o, \dots, x_k : A_k^o$. Consider an assignment \mathbf{a} and $b_1, \dots, b_k \in A$ such that $b_i \leq \llbracket A_i^o \rrbracket$. By hypothesis we get:

$$\begin{aligned} p\{x_1 := b_1, \dots, x_k := b_k\} &\leq \llbracket A^o \rrbracket \rightarrow \llbracket B^o \rrbracket \\ &\text{and} \\ q\{x_1 := b_1, \dots, x_k := b_k\} &\leq \llbracket A^o \rrbracket \end{aligned}$$

and by monotonicity of the application in \mathcal{A} we obtain:

$$pq\{x_1 := b_1, \dots, x_k := b_k\} \leq ([A^o] \rightarrow [B^o]) \quad [A^o] \leq [B^o]$$

Since the above is valid for all the assignments, we conclude that $\mathcal{A} \models \Gamma \vdash pq : [B^o]$.

For the quantifiers:

(\forall)_i Assume $\mathcal{A} \models \Gamma \vdash p : A^o$ and that x^σ does not appear free in Γ , where $\Gamma = x_1 : A_1^o, \dots, x_k : A_k^o$. Consider an assignment \mathfrak{a} and $b_1, \dots, b_k \in A$ such that $b_i \leq [A_i^o]$.

Since A_1^o, \dots, A_k^o does not depend upon x^σ , by the assumption $\mathcal{A} \models \Gamma \vdash p : A^o$, we get:

$$p\{x_1 := b_1, \dots, x_k := b_k\} \leq [A^o]\{x^\sigma := s\} \text{ for all } s \in [\sigma]$$

Then $p\{x_1 := b_1, \dots, x_k := b_k\} \leq \inf\{[A^o]\{x^\sigma := s\} \mid s \in [\sigma]\} = [(\forall x^\sigma A^o)^o]$. We conclude as before that $\mathcal{A} \models \Gamma \vdash p : (\forall x^\sigma A^o)^o$.

(\forall)_e Assume $\mathcal{A} \models \Gamma \vdash p : (\forall x^\sigma A^o)^o$, where $\Gamma = x_1 : A_1^o, \dots, x_k : A_k^o$. Consider an assignment \mathfrak{a} and $b_1, \dots, b_k \in A$ such that $b_i \leq [A_i^o]$. By the assumption $\mathcal{A} \models \Gamma \vdash p : (\forall x^\sigma A^o)^o$ we get:

$$p\{x_1 := b_1, \dots, x_k := b_k\} \leq [A^o]\{x^\sigma := s\} \text{ for all } s \in [\sigma]$$

Since $[M^\sigma] \in [\sigma]$ we obtain:

$$p\{x_1 := b_1, \dots, x_k := b_k\} \leq [A^o]\{x^\sigma := [M^\sigma]\} = [A^o\{x^\sigma := M^\sigma\}]$$

We conclude as before that $\mathcal{A} \models \Gamma \vdash p : A^o\{x^\sigma := M^\sigma\}$. □

The language of higher-order Peano arithmetic $-(\text{PA})^\omega$ is an instance of \mathcal{L}^ω where we distinguish a constant of kind I and two constants of expression 0^I and $\text{succ}^{I \rightarrow I}$.

Definition 6.5. For each kind σ we define the Leibniz equality $=_\sigma$ as follows:

$$x_1^\sigma =_\sigma x_2^\sigma := \forall y^{\sigma \rightarrow o} \left((y^{\sigma \rightarrow o} x_1^\sigma)^o \Rightarrow (y^{\sigma \rightarrow o} x_2^\sigma)^o \right)^o$$

The axioms of Peano arithmetic are equalities over the kind I , except for $\forall x^I ((\text{succ}^{I \rightarrow I} x^I =_I 0^I) \Rightarrow \perp)^o$ –which we abbreviate $\forall x^I (\text{succ}^{I \rightarrow I} x^I \neq 0^I)^o$ – and for the induction principle.

Definition 6.6. Fixed $\mathcal{A} \in \mathcal{K} \text{OCA}$, we say that $a \in A$ realizes a formula F^o if $a \leq [F^o]$. We write $a \Vdash_{\mathcal{A}} F^o$ for “ a realizes F^o ”, or simply as $a \Vdash F^o$, whenever it does not cause confusion.

The theory of \mathcal{A} is the set of closed formulae F^o such that there is an $a \in \Phi$ which realizes F^o . The theory of \mathcal{A} is denoted by $\text{th}(\mathcal{A})$.

In this presentation of Krivine’s realizability, the orthogonality is implicit in the implication \rightarrow that is part of the structure of the $\mathcal{K} \text{OCA}$.

Lemma 6.7. Let us consider an equality $M^\sigma =_\sigma N^\sigma$ such that $[M^\sigma] = [N^\sigma]$. Then the equality $M^\sigma =_\sigma N^\sigma$ is realized by $\mathbf{e}(\lambda^* x.x)$.

Proof. Consider an $f \in [\sigma \rightarrow o] = A^{[\sigma]}$, since $[M^\sigma] = [N^\sigma]$ we have $f([M^\sigma]) = f([N^\sigma])$. We conclude that $(\lambda^* x.x)[y^{\sigma \rightarrow o} M^\sigma] \leq [y^{\sigma \rightarrow o} M^\sigma] = [y^{\sigma \rightarrow o} N^\sigma]$ and $\mathbf{e}(\lambda^* x.x) \leq [y^{\sigma \rightarrow o} M^\sigma \Rightarrow y^{\sigma \rightarrow o} N^\sigma]$ for every assignment of $y^{\sigma \rightarrow o}$. Hence $\mathbf{e}(\lambda^* x.x) \Vdash M^\sigma =_\sigma N^\sigma$. □

Proposition 6.8. In every $\mathcal{K} \text{OCA}$ \mathcal{A} all axioms of Peano arithmetic but the induction principle are in $\text{th}(\mathcal{A})$.

Proof. By 6.7 all the axioms which are equalities are realized by $\mathbf{e}(\lambda^* x.x)$. Moreover, the axiom which say that 0 is not a successor is also realized: It is easy verify that $[\forall x^I [\text{succ}^{I \rightarrow I} x^I =_I 0^I \Rightarrow \perp]] = [\top \Rightarrow \perp] \rightarrow [\perp]$. By monotonicity $[\top \Rightarrow \perp]s \leq [\top \Rightarrow \perp][\top] \leq [\perp]$. Thus $[\top \Rightarrow \perp]s \leq [\perp]$ and hence $\mathbf{e}(\lambda^* x.xs) \Vdash [\forall x^I [\text{succ}^{I \rightarrow I} x^I =_I 0^I \Rightarrow \perp]]$ □

Definition 6.9. The formula $\mathbb{N}(z^I)$ is defined as:

$$\forall x^{I \rightarrow o} (\forall y^I ((x^{I \rightarrow o} y^I)^o \Rightarrow (x^{I \rightarrow o} (\text{succ}^{I \rightarrow I} y^I))^o \Rightarrow ((x^{I \rightarrow o} 0^I)^o \Rightarrow (x^{I \rightarrow o} z^I)^o)^o$$

Remark 6.10. Since the equational axioms of Peano arithmetic and the axiom $\forall x^I [\text{succ}^{I \rightarrow I} x^I =_I 0^I \Rightarrow \perp]$ are universal formulæ and, therefore, imply their relativization to \mathbb{N} . The relativization of the induction principle to \mathbb{N} is $\forall x^I (\mathbb{N}(x^I) \Rightarrow \mathbb{N}(x^I))$, which is realized by means of $\mathbf{e}(\lambda^* x.x)$. Thus, relativizing to \mathbb{N} all proofs of higher-order arithmetic, we find realizers in Φ for their theorems by means of adequacy 6.4. In other words, $\text{th}(\mathcal{A})$ contains $\text{th}((PA)^\omega)$.

REFERENCES

- [1] Friedman, H. *Some applications of Kleene's methods for intuitionistic system*, Cambridge summer school in mathematics, 337, pp 113-170 (1973)
- [2] Gierz, G., Hofmann, K. H., Keimel, K., Lawson, J.D., Mislove, M., Scott, D. S. *Continuous Lattices and Domains*. Cambridge University Press, 2003.
- [3] Griffin, T.G. *A Formulæ-as-Types Notion of Control*. In Conference Record of the Seventeenth Annual ACM Symposium on Principles of Programming Languages, 1990.
- [4] Hofstra, P. *All realizability is relative*, Math. Proc. Cambridge Philos. Soc. 141 (2006), no. 2, pp. 239–264.
- [5] Hofstra, P. *Iterated realizability as a comma construction*, Math. Proc. Cambridge Philos. Soc. 144 (2008), no. 1, pp. 39–51.
- [6] Hyland, J. M. E. *The effective topos*, Proc. of The L.E.J. Brouwer Centenary Symposium (Noordwijkerhout 1981) pp. 165-216, North Holland 1982
- [7] Hyland, J.M.E., Johnstone, P.T., Pitts, A.M. *Tripes theory*, Math. Proc. Cambridge Phil. Soc. 88 (1980), pp 205–232.
- [8] Johnstone, P.T. *Sketches of an elephant: a topos theory compendium. Vol. 1*, volume 43 of *Oxford Logic Guides*. The Clarendon Press Oxford University Press, New York, 2002.
- [9] Hofstra, P., van Oosten, J. *Ordered partial combinatory algebras*, Math. Proc. Cambridge Philos. Soc. 134 (2004), no. 3, pp. 445–463.
- [10] Kleene, S.C. *On the interpretation of intuitionistic number theory.*, Journal of Symbolic Logic, 10:109-124, 1945.
- [11] Krivine, J.L., *A general storage theorem for integers in call-by-name lambda-calculus.*, Th. Comp. Sc., 129, p. 79-94 (1994).
- [12] Krivine, J.-L., *Realizability algebras: a program to well order \mathbb{R}* , Logical methods in computer science, vol 7, (3: 02), 2001, pp 1–47.
- [13] Krivine, J.-L. *Types lambda-calculus in classical Zermelo-Fraenkel set theory*, Arch. Math. Log. 40 (2001), no. 3, pp. 189–205.
- [14] Krivine, J.-L., *Dependent choice, 'quote' and the clock*, Th. Comp. Sc. 308 (2003), pp. 259–276.
- [15] Krivine, J.-L. *Structures de réalisabilité, RAM et ultrafiltre sur \mathbb{N}* , (2008). [http : //www.pps.jussieu.fr/~krivine/Ultrafiltre.pdf](http://www.pps.jussieu.fr/~krivine/Ultrafiltre.pdf).
- [16] Krivine, J.-L. *Realizability in classical logic in Interactive models of computation and program behaviour*, Panoramas et synthèses 27 (2009), SMF.
- [17] Lack, S. *A 2-categories companion*. In *Towards higher categories*, pages 105–191. Springer, 2010.
- [18] Lawvere, F.W. *Adjointness in foundations*. *Dialectica*, 23(3-4):281–296, 1969.
- [19] Lawvere, F. W. *Equality in hyperdoctrines and the comprehension schema as an adjoint functor*. *Applications of Categorical Algebra*, 17:1–14, 1970.
- [20] McCarty, D. *Realizability and recursive mathematics*, Technical Report CMU-CS-84-131. Departament of Computer Science, Carnegie-Mellon University, 1984. Report version of the author's PhD thesis, Oxford University 1983.
- [21] Myhill, J. *Some properties of intuitionistic Zermelo-Fraenkel set theory*, Lecture notes in mathematics 337, pp 206-231 (1973).
- [22] Sørensen, M.H., Urzyczyn, P. *Lectures in the Curry-Howard isomorphism* Elsevier, Studies in Logic and the foundations of mathematics, vol 149, 2006.
- [23] Streicher, T. *Krivine's Classical Realizability from a Categorical Perspective*, Math. Struct. in Comp. Science
- [24] van Oosten, J. *Realizability, an Introduction to its Categorical Side*, (2008), Elsevier.